

Spring 5-1-2001

# A Fantasy Basketball Database

Martin Lundell  
*Dakota State University*

Follow this and additional works at: <https://scholar.dsu.edu/theses>

---

## Recommended Citation

Lundell, Martin, "A Fantasy Basketball Database" (2001). *Masters Theses*. 12.  
<https://scholar.dsu.edu/theses/12>

This Thesis is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses by an authorized administrator of Beadle Scholar. For more information, please contact [repository@dsu.edu](mailto:repository@dsu.edu).

***A Fantasy Basketball Database***

By Martin Lundell

A Program Project

Submitted in Partial Fulfillment  
of the Requirement for the Degree of  
Master of Science in Information Systems

Dakota State University

May 2001

Approved by:

\_\_\_\_\_

Dr. Ronghua Shan, Chair

\_\_\_\_\_

Dr. Terry Dennis

\_\_\_\_\_

Heidi Struve

**Fantasy Basketball Database** [REDACTED]

**Martin Lundell**  
INFS 790 Project  
Spring 2001

**Review Board Members:**

Dr. Ronghua Shan, Dakota State University, (Chair)

Dr. Terry Dennis, Dakota State University

Heidi Struve, [REDACTED]

## Table Of Contents

<b>I.</b>	<b>Introduction.....</b>	<b>3</b>
<b>II.</b>	<b>Scope.....</b>	<b>5</b>
<b>III.</b>	<b>Design Methology.....</b>	<b>7</b>
<b>IV.</b>	<b>Performance Issues.....</b>	<b>10</b>
<b>V.</b>	<b>DTS Packages.....</b>	<b>11</b>
<b>VI.</b>	<b>Data Integrity Issues.....</b>	<b>12</b>
<b>VII.</b>	<b>Conclusion.....</b>	<b>13</b>
<b>VIII.</b>	<b>Appendicies.....</b>	<b>14</b>
	<b>a. Project Schedule</b>	
	<b>b. Data Models</b>	
	<b>c. Test Plan</b>	
	<b>d. Metadata</b>	
	<b>e. Stored Procedures</b>	
	<b>f. Power Point Slides</b>	



## **I. Introduction**

This is a summary of the development of [REDACTED]'s fantasy basketball game. Although the development for this project consisted of both front-end web development and back-end database development, this paper will focus primarily on the database design and implementation as that was my role in the project.

[REDACTED] is a site where fantasy sports enthusiasts can play fantasy games online. Fantasy sports games are games of skill that pit an individual's knowledge of a particular sports organization (i.e. The NBA, NFL) against others competing in a common league. The basic premises for the games are that all persons playing in a fantasy league will select professional players from the corresponding organization's teams. If an individual's selected players score more points in a given period of time, they win.

Fantasy games can be set up in many ways, with many different rules. The Internet has facilitated the fantasy sports industry, bringing together fantasy players from all over the globe to compete against each other and receive results and prizes in real time.

[REDACTED] has never had a fantasy basketball game, although it did offer a fantasy football game. The football and basketball databases remained separate except in that they share the same users table residing in the football database so players can use their same username and login as they had in football as well as allowing them to use the same account for monetary transactions. They also share a table that contains all cobrand information so that sites can more easily be cobranded for both games. (Cobranded is a method of integrating one sites functionality into another site so that the functionality takes on the same look and feel of the other site.)

This is an example of a project, which unfortunately, is probably far too typical of many projects undertaken in the IT industry. A project of this size and scope should have had a life cycle of 8 to 10 months, not the 8 weeks in which it was completed. The time constraints in this project proved to be the biggest challenge especially in regards to testing and documentation.

The development team consisted of two Cold Fusion developers and one Database developer (the author of this paper). The development of this project began while contract negotiations with the client were still underway and finished just days before the start of the NBA regular season schedule with work weeks climbing into the 70-hour range. Testing of the project often happened side-by-side with development of functionality, compromising its validity as that functionality became integrated with new functionality being built.

That being said, the fantasy basketball project was completed on schedule, and while sparsely documented, is fully functional and automated.

Most of the data processing logic for this site was built into the database's stored procedures. [REDACTED] is database driven wherever possible so as to make the site dynamic and to allow for easy game administration. This philosophy was followed so much as to keep color schemes, and image paths for cobrand partners in the database so different color schemes and images would appear depending on which site you were accessing the game from.

## II. Scope

Project start date: 9/11/2000

Project completion date: 10/20/2000;

The fantasy basketball game will be in effect from Tuesday, October 31, 2000 until the final regular season game on or about Wednesday, April 18, 2001

Each starting lineup consists of 14 players who can be from any of the 29 pro basketball teams and whose total values do not exceed a \$50,000,000 value cap. Each team has:

5 Guards

5 Forwards

3 Centers

1 Swingman (Any Position)

A player's position and values, was assigned by the client, [REDACTED], and will remain the same for the entire 2000 - 2001 season.

Each team selects a bench consisting of 11 players. These players are backups for the starting team and may be used during the season to replace a starter for any reason. (Typically the bench is used to replace injured players and players that do not perform as expected.) The \$50,000,000 value cap and position distribution rules do not apply to the bench. If these players are moved to the starting team, however, the value cap and position rules do apply.

An individual may select players who are not listed, however, all unlisted players are assigned a \$1,000,000 value and must start the game on the bench. Unlisted players, if selected, cannot be placed in the starting lineup until Week 2 for games during the week of Monday, November 6, 2000 thru Sunday, November 12, 2000.

### **League Formation**

Leagues are formed on a first-come, first-serve basis. Twenty-five teams form a league and there is a limit of one team per player, per league.

Players who would like to be in the same league with friends/colleagues may reserve up to 10 spots in a league.

### **Transactions**

Individuals may modify their starting lineup via the Internet within the constraints of the \$50,000,000 salary cap, and their bench, as needed, for a nominal fee each week during the contest. Player transactions must be made via the Internet at

[REDACTED]

Transactions must be completed for the first week of the season on Tuesday, October 31, 2000 before 5 PM CDT. Transactions for the rest of the season must be completed before 11 PM CDT each Sunday in order to be effective for the following Monday thru Sunday games.

Any transactions made after a deadline do not take effect until the next deadline. A player's roster carries over through each deadline if they do not make any transactions.

There are three types of transactions: (1) bench switch, (2) swing move, and (3) new purchase.

A **bench switch** occurs when an individual replaces a player from their starting lineup with a player from their bench. Bench switches are unlimited during the contest and must involve players listed at the same position. Bench switches also include switching players who have a buy week from an individual's starting lineup to the bench. Bench switch example: "Start #101 Gary Payton, Bench #103 Kobe Bryant."

A **swing shift** occurs when an individual moves the swing player in their starting lineup to another position in their starting lineup. Swing shifts are unlimited during the Contest and must be accompanied by a bench switch and/or a new player purchase. Swing shift example: "Shift #317 Dirk Nowitzki to forward for #332 Anthony Mason. Start #116 Jason Kidd, Bench #332 Anthony Mason."

A **new purchase** occurs when an individual adds a new player to either their starting lineup or bench and drops a player from their roster. An individual may purchase any listed or unlisted player not on their current roster. New purchases are limited to 11 per roster during the contest. New purchase example: "Purchase #301 Shaquille O'Neal, Drop #302 Vince Carter."

All transactions (bench/swing switches and new purchases) are free and unlimited prior to the entry deadline (5 PM CDT, Tuesday, October 31, 2000).

### **Transaction Costs**

<b>Transaction</b>	<b>Pre-Season</b>	<b>Season</b>
Bench/Swing Moves	Free	\$1.00 (unlimited)
Trades/New Purchases	Free	\$5.00 (limit 11)

## **Scoring**

<b><u>Category</u></b>	<b><u>Point Value</u></b>
2 point field goals	(2)
3 point field goals	(3)
Free throw	(1)
Rebound	(2)
Assist	(1)
Steal	(3)
Block	(3)

Points are totaled each week for the players in an individual's starting line-up only, regardless to the position they actually play during the game. Points accumulated by players on the bench will not count towards the weekly and overall total.

The stats are accumulated for games during the week beginning on Monday and ending on Sunday.

Player statistics will be delivered by a vendor in a CSV text file via FTP on a daily basis. These statistics will be imported into the database and applied against FantasyChallenges.com's scoring rules. Players' scores and statistics will be reflected cumulatively throughout the week on each participant's roster by 11:00 AM CST each day.

Past weeks' rosters and the roster's scores and stats will be archived and available for review on the website at all times.

Statistics are accumulated during the regular season only.

Overall Standings are comprised of all teams in this Challenge. To determine standings, all teams in the Challenge and all teams in their respective leagues are sorted by point total. The team with the highest point total finishes first and the team with the lowest point total finishes last.

## **Cobranding**

[REDACTED] will be cobranded on sites across the Internet. Leagues may be required (based on individual cobrand agreements) to be formed based on the cobrand through which an individual bought a game. This may be required for advertising and cobrand-specific prizes.

### III. Design Methodology

From a conceptual standpoint there were only two user views to be concerned about, the user of the game and the administrator of the game. These views are conceptually the same, differing only in the scope of their transactions. A user of the game could only perform transactions relating to his roster, whereas an administrator was a super-user who could perform transactions for all users.

It was in this conceptual/logical stage where entities, their relationships, and their attributes were laid out. The modeling at this point reflected more of a global logical data model than a conceptual one as, right from the start, the model did not contain many-to-many relationships, complex relationships, relationships with attributes, multi-valued relationships or redundant relationships. The data model at this point was normalized to third normal form. There were no repeating groups, no partial dependencies on the primary keys, and no transitive dependencies on the primary keys.

To complete the logical design phase, the model was validated against anticipated user transactions. These transactions included: signing up as a user; buying a game; adding, dropping, benching, and activating players; reserving spots in leagues, scoring a roster, as well as a myriad of different queries to pull and display information for both the user and the administrator.

The physical database design was done in SQL Server 7.0 standard edition. (The choice for this RDBMS was based more on readily available resources than anything else.) SQL Server 7.0's Enterprise manager was used to create the tables, their primary key, and foreign key constraints.

During the physical database design we addressed potential issues regarding the functionality and frequency of user transactions on the database and made some changes accordingly. These changes mostly dealt with archiving certain information that would no longer require update, insert (except when archiving a large dataset), or deletes but rather would only be used in select statements for reporting purposes. Also certain tables were added to address certain transactional needs based on the importing of statistics.

SQL Server 7.0 uses B<sup>+</sup>-Tree indexing on tables and can only have one clustered index per table. This clustered index was almost always applied to the table's primary key. Secondary (non-clustered) indexes were not implemented, although there are a few columns that could benefit from such an index, specifically in the player table on the columns statsincid and bdsportsid as these are often referenced in the where clause of queries.

Security for the database is as follows: There will physically be only one user accessing this database from the web front-end. That user is the one that the Cold Fusion Administrator uses when someone logs on the website. This user has no rights specified in the RDBMS except that it owns the database for this web site. By having ownership it has all DML and DDL rights needed to query, update, create and delete tables and to execute stored procedures that also do those same things. The only other login that has access to this database is the administrator for this instance of SQL Server.

## **Performance Issues**

This site, by the nature of the game being played here, has a heavy amount of data transfers and transactions. With virtually every page having to do multiple queries to the database, the speed of the site quickly became an issue. To address this issue the vast majority of database transactions were done through stored procedures where the DBMS stores the complied execution plan of the stored procedure rather than recreating an execution plan from the front-end code each time the page is accessed.

A decision was made to use separate tables for current roster information and old roster information even though the tables are structurally identical. The old roster information archives are used for reporting purposes and are accessed frequently. The current roster information is also frequently displayed but is also heavily used in transactions. For performance reasons we felt it would be best to keep the current and old information separate. An example of a performance related issue would be that of contention locking when multiple users are attempting to access the same information.



## **DTS (data transformation services) Packages**

One of the major considerations from the database development side was the automation or importing and computing player statistics. Files of player statistics were provided by an outside-vendor on a daily basis (by 7:00 AM) for players who played in the previous day's games. It was arranged with the vendor that they would FTP data in a CSV (comma separated value) file to a specified directory on our FTP server. A DTS package was created that contained a call to a batch file that downloaded that CSV file into a directory on the IIS server. This CSV file would overwrite the file from the previous day. Upon successful completion of this step the DTS package would import the data from the CSV file into a table in the database. Once the data was in the database, the DTS package would kick off stored procedures and SQL statements to manipulate and process that data into scores and statistics for the user's current rosters. We then used SQL Server's ability to integrate with the Windows NT operating system to schedule that package as a job that would run daily at 11:00 AM.

A potential risk to this process is the failure of timely delivery of the statistics file from the vendor. The DTS package at this point in time does not know if the file is the current days file or the previous day's file. There exists the potential that the DTS package would import the same file two (or more) days in a row if the vendor was late in their file upload to the directory on the FTP server. The proper checks could be developed within the application to ensure that this does not happen, but available resources and certain political considerations have prevented this from happening.

Two other DTS packages were needed in order to fully automate this game. One package would, on a weekly basis, archive the previous week's roster. The previous week's roster is the roster the users have in place for the upcoming week's games. The current roster table holds the roster that the user is building for next week's games. The last DTS package is one that archives the statistics from the vendor and clears out the import table for the upcoming week.

## **Data Integrity**

Several data integrity issues needed to be addressed. One was that of unique identifiers for players. The client who was building the game had a unique identifier that they used for ranking a player and assigning a dollar value based on that ranking. They also used that identifier in literature to customers playing the game. So from a game administrative vantage point, they wanted to be able to use their identifiers for all player transactions in the game. The problem here lies in the fact that each year, their numbers will change based on the performance of the player in the previous year. So it would be irresponsible from a development standpoint to use their numbering system. The statistics vendor also had an identifier for the players, which they guaranteed to be unique. There was concern, however, that they may decide to go with a different vendor for statistics in the future, and consequently did not want to have any data issues if they decided to do so. To resolve these issues, a third unique identifier was created by the development team to serve as a switching point. Before the start of the season we obtained a list of players with their unique identifiers from both the client and the vendor and put them in the Player table while assigning our own unique identifier. All player statistics from the vendor are resolved by matching up their ids in the player table while game transactions from the client are resolved against their ids in the player table. But the unique identifier that resides in the customer's rosters, are the ones we assigned.

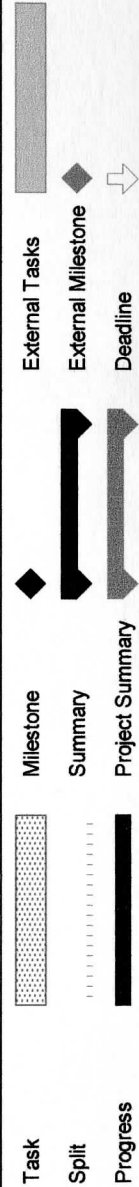
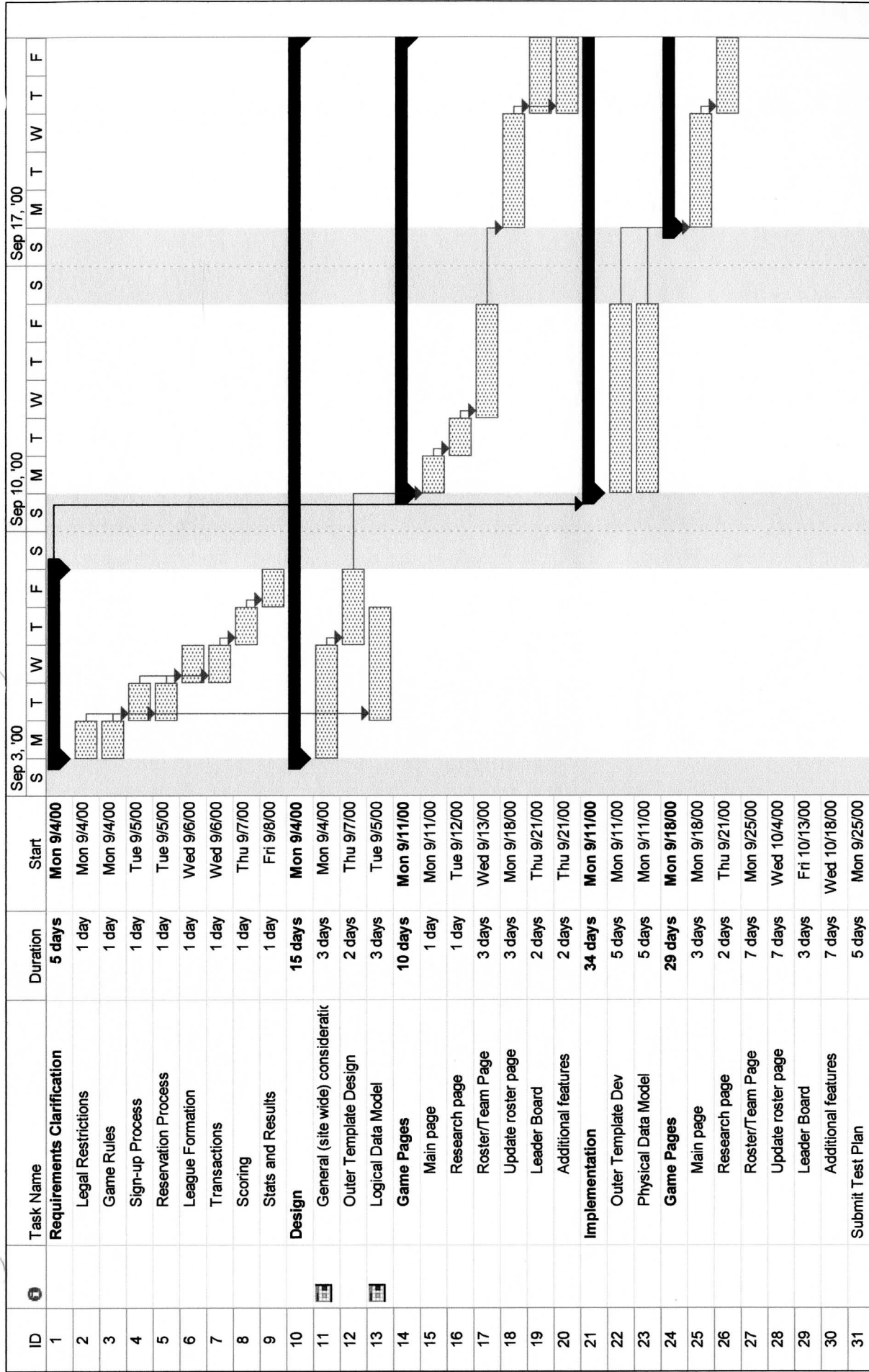
Other data integrity issues addressed revolved around processing transactions (Add, Drop, Bench, Swing) on rosters. There are several situations where transactions should not be processed. For example, if their credit card was canceled, the transaction was canceled by the user, or the transactions would violate the rules of the games in regards to trade or salary cap limits. In these situations we want to revert to the roster that was in place before the transaction was initiated. In order to do this we implemented a working table called RosterTemp.

When a user accesses the update roster page, their roster gets copied into the RosterTemp table. All user transactions are performed against their roster in that table and, if everything is approved, the roster gets copied back to the current Roster table.

## Conclusion

The conclusion is that the fantasy basketball game for [REDACTED] was completed on schedule, fulfilling scope requirements. It was a challenging project primarily because of the time constraints it needed to be developed in. Certain functionality of the site could be running more efficiently had there been more time to test and tune. Requirements concerning performance were not included in the scope for this site. The owners of [REDACTED] were looking for functionality first, to get the game into production and start receiving revenue. There is no contract in place for maintenance or site enhancements at this point, although negotiations have started. An issue that needs to be addressed is documentation of stored procedure code, DTS packages and Cold Fusion code. All have very little documentation, and neither the client or [REDACTED] are prepared to pay for developers time to finish that documentation. This will be an issue addressed in current negotiations on maintenance and site enhancements.

**APPENDIX A**



Project: FBB\_WBSandProjectPlan  
Date: Mon 4/16/01

ID	Task Name	Duration	Start	Sep 3, '00							Sep 10, '00							Sep 17, '00						
				S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S
32	Testing	20 days	Mon 10/2/00																					
33	General Layout	2 days	Mon 10/2/00																					
34	Game Pages	2 days	Mon 10/2/00																					
35	User Acct	3 days	Wed 10/4/00																					
36	Game Purchase Process	5 days	Mon 10/9/00																					
37	League Functionality	5 days	Mon 10/16/00																					
38	Roster Functionality	5 days	Mon 10/16/00																					
39	Game Research	2 days	Mon 10/16/00																					
40	Score Processing	5 days	Mon 10/23/00																					
41	Co-Brand Implementation	20 days	Mon 10/2/00																					

Project: FBB\_WBSandProjectPlan

Date: Mon 4/16/01

Task

Split

Progress

Milestone

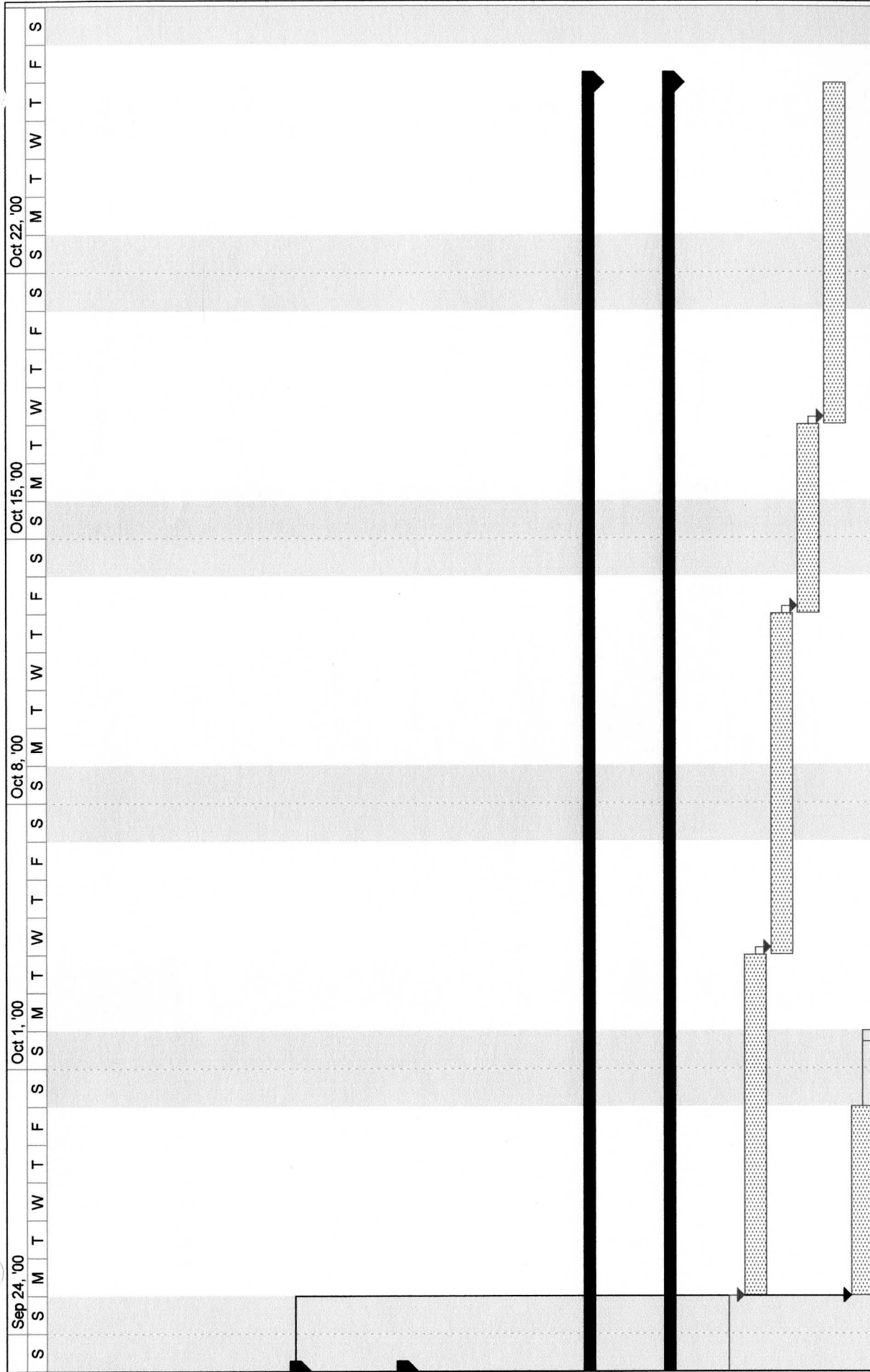
Summary

Project Summary

External Tasks

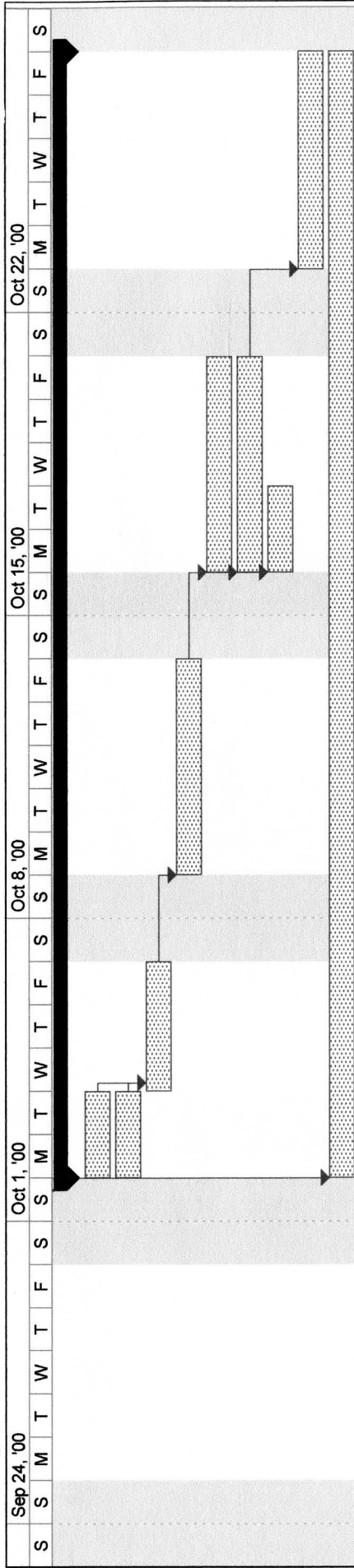
External Milestone










Deadline



<p>Project: FBB_WBSandProjectPlan Date: Mon 4/16/01</p>	<p>Task Split Progress</p>	<p>Milestone Summary Project Summary</p>	<p>External Tasks External Milestone Deadline</p>
---	------------------------------------	--	---

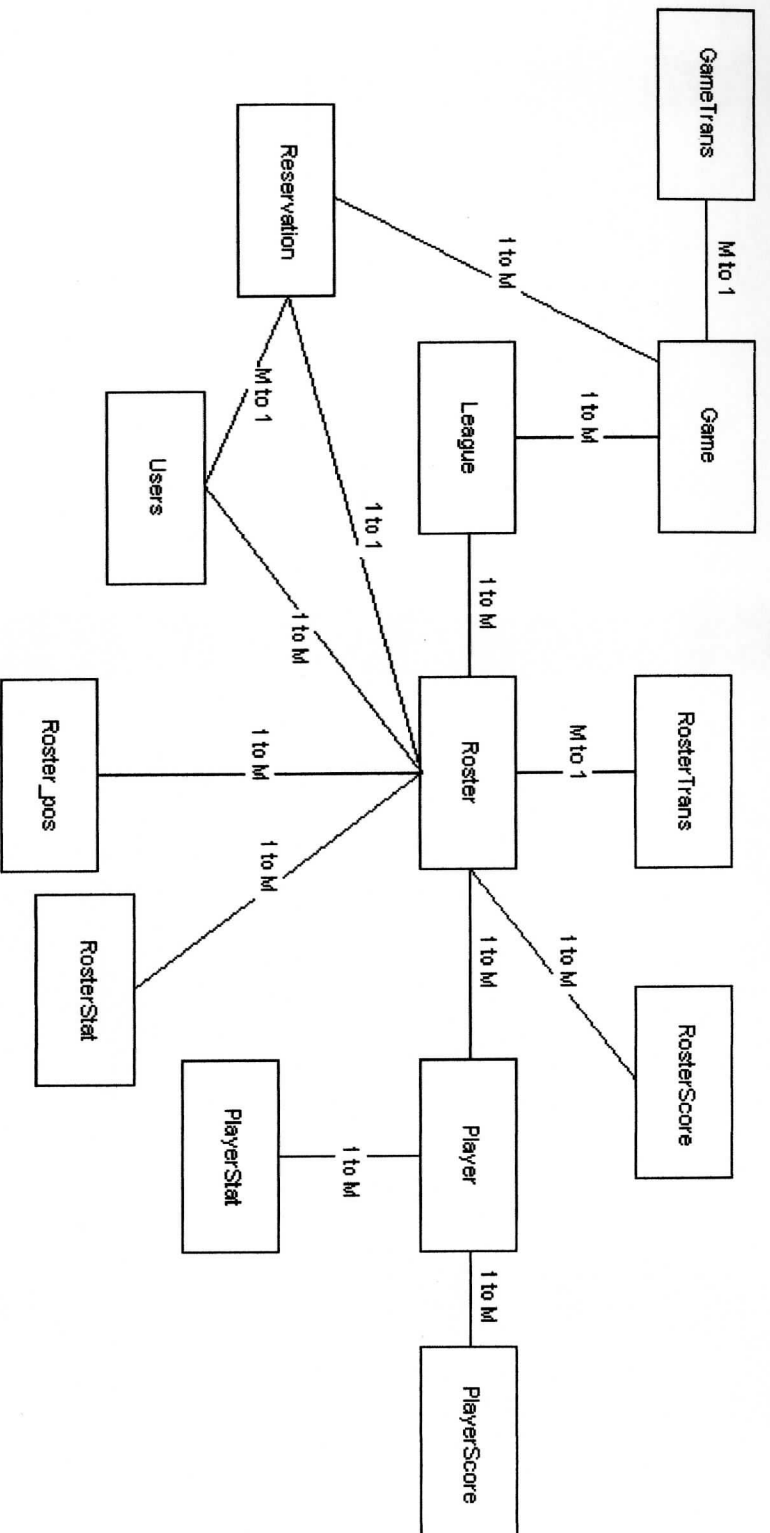




Project: FBB_WBSandProjectPlan Date: Mon 4/16/01	
Task	
Split	
Progress	
Milestone	
Summary	
Project Summary	
External Tasks	
External Milestone	
Deadline	



## **APPENDIX B**





## **APPENDIX C**

Area of the Site	IE 4.0		IE 5.0		IE 5.5		NS 4.0		NS 4.5		NS 4.75	
General Layout	P	F	P	F	P	F	P	F	P	F	P	F
Tables												
Color Schemes												
Colors												
Game Purchase Process												
Create Team												
Reservations												
Discount Codes												
Official Rules												
Credit Card Processing												
Roster Functionality												
Sortable/Viewable Stats												
Player Profiles												
Clubhouse Info												
Trades Remaining												
Standings												
Salary Total												
Total Points												
Active Roster Moves												
Add												
Drop												
Bench												
Activate												
Swing												
Bench Roster Moves												
Add												
Drop												
Bench												
Activate												
Swing												
Submit Changes												
Credit Card Processing												
League Functionality												
Overall Standings												
League Standings												
Viewable Rosters												
Sortable Stats												
Game Pages												
Rules												
Prizes												
Sign Up												
Game Home												
FAQ												
Research												
Schedule												
Chat												
Contact												
User Account												
Account Creation												
Forgot Password												
Update Profile												
Transaction History												
Account Balance												
Game Research												
Injury Report												
Players												
Schedules												
Score Processing												
Stats Inc. File Imports												
Injury												
Player												
Roster Score Updates												
Roster Stats Updates												
Player Score Updates												
Player Stats Updates												
Conversion of Stats to Scores					</							

## **APPENDIX D**

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PREC: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
drp	droptid	int identity	10	4	0 NULL	NO	uniquely identifies drp
drp	rosterid	int	10	4	0 NULL	NO	roster player is dropped from
drp	btgid	int	10	4	0 NULL	NO	BTG's ID of player dropped
drp	droptdate	datetime	23	16	3 NULL	YES	date Player was dropped

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PREC: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
game	gameid	int identity	10	4	0 NULL	NO	Uniquely identifies Game
game	game	nvarchar	50	100 NULL	NULL	NO	Name of Game
game	startdate	datetime	23	16	3 NULL	NO	Game start date
game	enddate	datetime	23	16	3 NULL	NO	Game end date
game	regdeadline1	datetime	23	16	3 NULL	NO	Deadline for early game registration
game	regdeadline2	datetime	23	16	3 NULL	YES	Deadline for game registration
game	imagepath	varchar	50	100 NULL	NULL	NO	Path to game's image
game	largeimage	varchar	50	100 NULL	NULL	YES	Game's description
game	description	varchar	1400	2800 NULL	NULL	NO	Game's description
game	free	bit	1	1	0 NULL	NO	flag for whether it is a free or paid game
game	livegame	bit	1	1	0 NULL	NO	flag for when game is active
game	href	varchar	255	255 NULL	NULL	YES	link to game home
game	root	varchar	255	255 NULL	NULL	YES	holds variable for game's root directory
game	ruleshref	varchar	255	255 NULL	NULL	YES	link to game's rule's page
game	clubimage	varchar	255	255 NULL	NULL	YES	Path to game's clubhouse image
game	lockoutstarttime	datetime	23	16	3 NULL	YES	time to implement lockout for player transactions
game	lockoutendtime	datetime	23	16	3 NULL	YES	time to remove lock on player transactions
game	lockoutstartday	int	10	4	0 NULL	YES	day on which lockout time is to happen on
game	lockoutendday	int	10	4	0 NULL	YES	day on which lockout time is to end on

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PREC: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
GamePurch	Userid	int	10	4	0 NULL	NO	Uniquely identifies a user
GamePurch	Gameid	int	10	4	0 NULL	NO	Identifies which game a user purchased

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
GameTrans	Gameid	int	10	4	0 NULL	NO	Uniquely identifies a game
	TransactionTypeid	int	10	4	0 NULL	NO	Uniquely identifies a transaction type
	Cost	decimal	18	20	2 NULL	YES	Dollar amt for transaction for specific game

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
ImportInjury	FName	varchar	255	255 NULL	NULL	YES	Player first name
	LName	varchar	255	255 NULL	NULL	YES	Player last name
	ID#	int	10	4	0 NULL	NO	Unique ID from stats vendor
	Tm	char	3	3 NULL	NULL	YES	Player's professional team abbr
	T#	int	10	4	0 NULL	YES	Vendor's Unique TeamID
	Date	datetime	23	16	3 NULL	NO	Date of Injury
ImportInjury	Description	varchar	255	255 NULL	NULL	YES	Desc. Of Injury

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
ImportPlayerStats	FName	varchar	255	255 NULL	NULL	YES	Player first name
	LName	varchar	255	255 NULL	NULL	YES	Player last name
ImportPlayerStats	ID#	int	10	4	0	NO	Unique ID from stats vendor
ImportPlayerStats	Tm	char	3	3 NULL	NULL	YES	Player's professional team abbr
ImportPlayerStats	T#	int	10	4	0 NULL	YES	Vendor's Unique TeamID
ImportPlayerStats	Ps	int	10	4	0 NULL	YES	Unused field from vendor
ImportPlayerStats	G	int	10	4	0 NULL	YES	Unused field from vendor
ImportPlayerStats	Min	int	10	4	0 NULL	YES	Total minutes player was in the game
ImportPlayerStats	Pts	int	10	4	0 NULL	YES	Total points player scored
ImportPlayerStats	FGM	int	10	4	0 NULL	YES	Field goals made
ImportPlayerStats	FGA	int	10	4	0 NULL	YES	Field goals attempted
ImportPlayerStats	FTM	int	10	4	0 NULL	YES	Freethrows made
ImportPlayerStats	FTA	int	10	4	0 NULL	YES	Freethrows attempted
ImportPlayerStats	3GM	int	10	4	0 NULL	YES	3 point Field goals made
ImportPlayerStats	3GA	int	10	4	0 NULL	YES	3 point field goals attempted
ImportPlayerStats	OReb	int	10	4	0 NULL	YES	Number of offensive rebounds
ImportPlayerStats	DReb	int	10	4	0 NULL	YES	Number of defensive rebounds
ImportPlayerStats	Reb	int	10	4	0 NULL	YES	Total Rebounds
ImportPlayerStats	Asst	int	10	4	0 NULL	YES	Total assists
ImportPlayerStats	St	int	10	4	0 NULL	YES	Total steals
ImportPlayerStats	Blk	int	10	4	0 NULL	YES	Total Blocks
ImportPlayerStats	TO	int	10	4	0 NULL	YES	Total turn overs
ImportPlayerStats	PF	int	10	4	0 NULL	YES	Total number of personal fouls
ImportPlayerStats	Tech	int	10	4	0 NULL	YES	Total number of technical fouls
ImportPlayerStats	ImportDate	datetime	23	16	3 (1 / 1 / 1900)	NO	Date and Time stats were imported into table
ImportPlayerStats	wk	int	10	4	0	NO	Week of season stats were imported into table



<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PREC: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
ImportPlayerStats_arc	FName	varchar	255	255 NULL	NULL	YES	Player first name
ImportPlayerStats_arc	LName	varchar	255	255 NULL	NULL	YES	Player last name
ImportPlayerStats_arc	ID#	int	10	4	0 NULL	NO	Unique ID from stats vendor
ImportPlayerStats_arc	Tm	char	3	3 NULL	NULL	YES	Player's professional team abbr
ImportPlayerStats_arc	T#	int	10	4	0 NULL	YES	Vendor's Unique TeamID
ImportPlayerStats_arc	Ps	int	10	4	0 NULL	YES	Unused field from vendor
ImportPlayerStats_arc	G	int	10	4	0 NULL	YES	Unused field from vendor
ImportPlayerStats_arc	Min	int	10	4	0 NULL	YES	Total minutes player was in the game
ImportPlayerStats_arc	Pts	int	10	4	0 NULL	YES	Total points player scored
ImportPlayerStats_arc	FGM	int	10	4	0 NULL	YES	Field goals made
ImportPlayerStats_arc	FGA	int	10	4	0 NULL	YES	Field goals attempted
ImportPlayerStats_arc	FTM	int	10	4	0 NULL	YES	Freethrows made
ImportPlayerStats_arc	FTA	int	10	4	0 NULL	YES	Freethrows attempted
ImportPlayerStats_arc	3GM	int	10	4	0 NULL	YES	3 point field goals made
ImportPlayerStats_arc	3GA	int	10	4	0 NULL	YES	3 point field goals attempted
ImportPlayerStats_arc	OReb	int	10	4	0 NULL	YES	Number of offensive rebounds
ImportPlayerStats_arc	DReb	int	10	4	0 NULL	YES	Number of defensive rebounds
ImportPlayerStats_arc	Reb	int	10	4	0 NULL	YES	Total Rebounds
ImportPlayerStats_arc	Asst	int	10	4	0 NULL	YES	Total assists
ImportPlayerStats_arc	St	int	10	4	0 NULL	YES	Total steals
ImportPlayerStats_arc	Blk	int	10	4	0 NULL	YES	Total Blocks
ImportPlayerStats_arc	TO	int	10	4	0 NULL	YES	Total turn overs
ImportPlayerStats_arc	PF	int	10	4	0 NULL	YES	Total number of personal fouls
ImportPlayerStats_arc	Tech	int	10	4	0 NULL	YES	Total number of technical fouls
ImportPlayerStats_arc	ImportDate	datetime	23	16	3 NULL	NO	Date and Time stats were imported into table
ImportPlayerStats_arc	wk	int	10	4	0 NULL	NO	Week of season stats were imported into table

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PREC: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
league	leagueid	int identity	10	4	0 NULL	NO	Uniquely identifies a league
league	gameid	int	10	4	0 NULL	NO	ID of game this league is in
league	leaguename	varchar	255	255 NULL	NULL	YES	name of league
league	commissionerid	int	10	4	0 NULL	NO	ID of Commissioner of league
league	availablespots	int	10	4	0 NULL	NO	Number of available spots left for teams in this league
league	private	int	10	4	0 NULL	YES	not being used
league	active	int	10	4	0 NULL	YES	not being used
league	cobrandid	int	10	4	0 NULL	YES	ID of cobrand site league was signed up from
league	internal	bit	1	1	0	NO	flag for whether to include league's teams in official standings for prizes

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PREC:</u>	<u>LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
player	bigid	int identity	10	4	0	NULL	NO	Our unique id for players
	bdsportsid	int	10	4	0	NULL	YES	ID assigned by client for ranking players by previous years' performance
player	teamid	int	10	4	0	NULL	YES	Our unique ID for Professional Team a player plays for
player	positionid	int	10	4	0	NULL	YES	Unique id for player's position
player	playerstatusid	int	10	4	0	NULL	YES	Unique id for player's status (active, retired etc)
player	statsincid	int	10	4	0	NULL	YES	Vendor's unique id
player	fname	varchar	50	50	NULL	NULL	NO	Player first name
player	lname	varchar	50	50	NULL	NULL	NO	Player last name
player	salary	decimal	18	20	0	NULL	YES	Dollar value assigned to player
player	listed	bit	1	1	0	NULL	YES	Flag for whether player is listed or not
player	pos	char	3	3	NULL	NULL	YES	Player's official professional positions (as determined by prof organization)

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PREC:</u>	<u>LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
playerscore	statsincid	int	10	4	0	NULL	NO	Vendor's unique id for player
playerscore	wk	int	10	4	0	NULL	NO	Week of season
playerscore	fg	int	10	4	0	0	NO	Field goal points
playerscore	ft	int	10	4	0	0	NO	Free throw points
playerscore	fg3	int	10	4	0	0	NO	3 point field goal points
playerscore	rebounds	int	10	4	0	0	NO	Rebounds converted to points
playerscore	assists	int	10	4	0	0	NO	Assists converted to points
playerscore	steals	int	10	4	0	0	NO	Steals converted to points
playerscore	blocks	int	10	4	0	0	NO	Blocks converted to points

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PREC:</u>	<u>LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
playerscore_arc	statsincid	int	10	4	0	NULL	NO	Vendor's unique id for player
playerscore_arc	wk	int	10	4	0	NULL	NO	Week of season
playerscore_arc	fg	int	10	4	0	NULL	NO	Field goal points
playerscore_arc	ft	int	10	4	0	NULL	NO	Free throw points
playerscore_arc	fg3	int	10	4	0	NULL	NO	3 point field goal points
playerscore_arc	rebounds	int	10	4	0	NULL	NO	Rebounds converted to points
playerscore_arc	assists	int	10	4	0	NULL	NO	Assists converted to points
playerscore_arc	steals	int	10	4	0	NULL	NO	Steals converted to points
playerscore_arc	blocks	int	10	4	0	NULL	NO	Blocks converted to points

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
playerstat	statstincid	int	10	4	0 NULL	NO	Vendor's unique id for player
playerstat	wk	int	10	4	0 NULL	NO	Week of season
playerstat	fg	int	10	4	0	NO	Number of Field goals
playerstat	ft	int	10	4	0	NO	Number of Free throws
playerstat	fg3	int	10	4	0	NO	Number of 3 point field goals
playerstat	rebounds	int	10	4	0	NO	Number of Rebounds
playerstat	assists	int	10	4	0	NO	Number of Assists
playerstat	steals	int	10	4	0	NO	Number of Steals
playerstat	blocks	int	10	4	0	NO	Number of Blocks

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
playerstat_arc	statstincid	int	10	4	0 NULL	NO	Vendor's unique id for player
playerstat_arc	wk	int	10	4	0 NULL	NO	Week of season
playerstat_arc	fg	int	10	4	0 NULL	NO	Number of Field goals
playerstat_arc	ft	int	10	4	0 NULL	NO	Number of Free throws
playerstat_arc	fg3	int	10	4	0 NULL	NO	Number of 3 point field goals
playerstat_arc	rebounds	int	10	4	0 NULL	NO	Number of Rebounds
playerstat_arc	assists	int	10	4	0 NULL	NO	Number of Assists
playerstat_arc	steals	int	10	4	0 NULL	NO	Number of Steals
playerstat_arc	blocks	int	10	4	0 NULL	NO	Number of Blocks

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
playerstatus	playerstatusid	int	10	4	0 NULL	NO	Uniquely identifies status
playerstatus	status	varchar	50	50 NULL	NULL	NO	Description of status (active, retired etc.)

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
poll	pollid	int	10	4	0 NULL	NO	Uniquely identifies a Poll
poll	pollname	varchar	50	50 NULL	NULL	YES	Name of Poll
poll	question	varchar	250	250 NULL	NULL	YES	Poll Question
poll	polltype	int	10	4	0 NULL	YES	ID of Poll Type
poll	response1	varchar	50	50 NULL	NULL	YES	Potential answer to poll question
poll	response2	varchar	50	50 NULL	NULL	YES	Potential answer to poll question
poll	response3	varchar	50	50 NULL	NULL	YES	Potential answer to poll question
poll	response4	varchar	50	50 NULL	NULL	YES	Potential answer to poll question
poll	response5	varchar	50	50 NULL	NULL	YES	Potential answer to poll question
poll	r1count	int	10	4	0 NULL	NO	Number of times response one was selected
poll	r2count	int	10	4	0 NULL	NO	Number of times response two was selected
poll	r3count	int	10	4	0 NULL	NO	Number of times response three was selected
poll	r4count	int	10	4	0 NULL	NO	Number of times response four was selected
poll	r5count	int	10	4	0 NULL	NO	Number of times response five was selected
poll	totalvotes	int	10	4	0 NULL	NO	Total number of responses
poll	startdate	datetime	23	16	3 NULL	YES	poll start date
poll	enddate	datetime	23	16	3 NULL	YES	poll end date
poll	currentpoll	bit	1	1	0 NULL	NO	flags active polls

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS</u>	<u>LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
position	positionid	int	10	4	0	NULL	NO	Uniquely identifies a position
position	positionabbr	varchar	4	4	NULL	NULL	NO	Position abbreviation
position	positionname	varchar	50	50	NULL	NULL	NO	Position unabbreviated

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS</u>	<u>LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
reservation	reservationid	int	10	4	0	NULL	NO	Uniquely identifies a reservation
reservation	rosterid	int	10	4	0	NULL	NO	ID of Roster reserved
reservation	status	int	10	4	0	NULL	NO	Stats of whether reservation was accepted
reservation	email	varchar	50	50	NULL	NULL	NO	email of individual for whom a roster was reserved for
reservation	reserveremail	varchar	50	50	NULL	NULL	YES	email of individual who reserved the roster
reservation	gameid	int	10	4	0	NULL	YES	ID of Game roster was reserved for
reservation	resdate	datetime	23	16	3	NULL	YES	date reservation was made
reservation	reserverid	int	10	4	0	NULL	YES	User ID of individual who made the reservation

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS</u>	<u>LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
roster	rosterid	int	10	4	0	NULL	NO	Uniquely identifies a roster
roster	leagueid	int	10	4	0	NULL	NO	ID of league roster is in
roster	userid	int	10	4	0	NULL	YES	ID of user roster belongs to
roster	reservation	bit	1	1	0	0	NO	flag for whether this roster had been reserved
roster	teamname	varchar	50	50	NULL	NULL	YES	Name user gives to roster (team)
roster	trades	int	10	4	0	-11	NO	number of trades left for this roster
roster	salarysum	int	10	4	0	0	NO	current salary total of starting lineup in roster
roster	wk	int	10	4	0	0	NO	current week of season for this roster
roster	G1	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	G2	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	G3	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	G4	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	G5	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	F1	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	F2	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	F3	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	F4	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	F5	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	C1	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	C2	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	C3	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	SW	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	B1	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	B2	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	B3	int	10	4	0	0	NO	BTG ID of player in this position for this week

roster	B4	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	B5	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	B6	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	B7	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	B8	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	B9	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	B10	int	10	4	0	0	NO	BTG ID of player in this position for this week
roster	B11	int	10	4	0	0	NO	BTG ID of player in this position for this week

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
roster_arc	rosterid	int	10	4	0 NULL	NO	Uniquely identifies a roster
roster_arc	leagueid	int	10	4	0 NULL	NO	ID of league roster is in
roster_arc	userid	int	10	4	0 NULL	YES	ID of user roster belongs to
roster_arc	reservation	bit	1	1	0 NULL	NO	flag for whether this roster had been reserved
roster_arc	teamname	varchar	50	50 NULL	NULL	YES	Name user gives to roster (team)
roster_arc	trades	int	10	4	0 NULL	NO	number of trades left for this roster
roster_arc	salarysum	int	10	4	0 NULL	NO	current salary total of starting lineup in roster
roster_arc	wk	int	10	4	0 NULL	NO	current week of season for this roster
roster_arc	G1	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	G2	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	G3	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	G4	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	G5	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	F1	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	F2	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	F3	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	F4	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	F5	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	C1	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	C2	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	C3	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	SW	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	B1	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	B2	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	B3	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	B4	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	B5	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	B6	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	B7	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	B8	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	B9	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	B10	int	10	4	0 NULL	NO	BTG ID of player in this position for this week
roster_arc	B11	int	10	4	0 NULL	NO	BTG ID of player in this position for this week

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS</u>	<u>LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
roster_previousweek	rosterid	int	10			0 NULL	NO	Uniquely identifies a roster
roster_previousweek	leagueid	int	10			0 NULL	NO	ID of league roster is in
roster_previousweek	userid	int	10			0 NULL	YES	ID of user roster belongs to
roster_previousweek	reservation	bit	1			0 NULL	NO	flag for whether this roster had been reserved
roster_previousweek	teamname	varchar	50		50 NULL	NULL	YES	Name user gives to roster (team)
roster_previousweek	trades	int	10			0 NULL	NO	number of trades left for this roster
roster_previousweek	salarysum	int	10			0 NULL	NO	current salary total of starting lineup in roster
roster_previousweek	wk	int	10			0 NULL	NO	current week of season for this roster
roster_previousweek	G1	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	G2	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	G3	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	G4	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	G5	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	F1	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	F2	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	F3	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	F4	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	F5	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	C1	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	C2	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	C3	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	SW	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	B1	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	B2	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	B3	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	B4	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	B5	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	B6	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	B7	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	B8	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	B9	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	B10	int	10			0 NULL	NO	BTG ID of player in this position for this week
roster_previousweek	B11	int	10			0 NULL	NO	BTG ID of player in this position for this week

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
rosterscore	rosterid	int	10	4	0 NULL	NO	unique id for roster
rosterscore	wk	int	10	4	0 NULL	NO	Week of season
rosterscore	fg	int	10	4	0	YES	Roster field goal points
rosterscore	ft	int	10	4	0	YES	Roster Free throw points
rosterscore	fg3	int	10	4	0	YES	Roster 3 point field goal points
rosterscore	rebounds	int	10	4	0	YES	Roster Rebounds converted to points
rosterscore	assists	int	10	4	0	YES	Roster Assists converted to points
rosterscore	steals	int	10	4	0	YES	Roster Steals converted to points
rosterscore	blocks	int	10	4	0	YES	Roster Blocks converted to points
rosterscore	wktotal	int	10	4	0	YES	Roster's total points for this week

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
rosterscore_arc	rosterid	int	10	4	0 NULL	NO	unique id for roster
rosterscore_arc	wk	int	10	4	0 NULL	NO	Week of season
rosterscore_arc	fg	int	10	4	0 NULL	YES	Roster field goal points
rosterscore_arc	ft	int	10	4	0 NULL	YES	Roster Free throw points
rosterscore_arc	fg3	int	10	4	0 NULL	YES	Roster 3 point field goal points
rosterscore_arc	rebounds	int	10	4	0 NULL	YES	Roster Rebounds converted to points
rosterscore_arc	assists	int	10	4	0 NULL	YES	Roster Assists converted to points
rosterscore_arc	steals	int	10	4	0 NULL	YES	Roster Steals converted to points
rosterscore_arc	blocks	int	10	4	0 NULL	YES	Roster Blocks converted to points
rosterscore_arc	wktotal	int	10	4	0 NULL	YES	Roster's total points for this week

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
rosterstat	rosterid	int	10	4	0 NULL	NO	unique id for roster
rosterstat	wk	int	10	4	0 NULL	NO	Week of season
rosterstat	fg	int	10	4	0	YES	Number of field goals for roster
rosterstat	ft	int	10	4	0	YES	Number of free throws for roster
rosterstat	fg3	int	10	4	0	YES	Number of 3 point field goals for roster
rosterstat	rebounds	int	10	4	0	YES	Number of rebounds for roster
rosterstat	assists	int	10	4	0	YES	number of assists for roster
rosterstat	steals	int	10	4	0	YES	Number of steals for roster
rosterstat	blocks	int	10	4	0	YES	Number of blocks for roster

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS: LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
rosterstat_arc	rosterid	int	10	4	0 NULL	NO	unique id for roster
rosterstat_arc	wk	int	10	4	0 NULL	NO	Week of season
rosterstat_arc	fg	int	10	4	0 NULL	YES	Number of field goals for roster
rosterstat_arc	ft	int	10	4	0 NULL	YES	Number of free throws for roster
rosterstat_arc	fg3	int	10	4	0 NULL	YES	Number of 3 point field goals for roster
rosterstat_arc	rebounds	int	10	4	0 NULL	YES	Number of rebounds for roster
rosterstat_arc	assists	int	10	4	0 NULL	YES	number of assists for roster
rosterstat_arc	steals	int	10	4	0 NULL	YES	Number of steals for roster
rosterstat_arc	blocks	int	10	4	0 NULL	YES	Number of blocks for roster

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
rosteremp	rosterid	int	10	4	0 NULL	NO	Uniquely identifies a roster
rosteremp	LeagueID	int	10	4	0 NULL	YES	ID of league roster is in
rosteremp	UserID	int	10	4	0 NULL	YES	ID of user roster belongs to
rosteremp	Reservation	bit	1	1	0	NO	flag for whether this roster had been reserved
rosteremp	TeamName	varchar	50	50 NULL	NULL	YES	Name user gives to roster (team)
rosteremp	Trades	int	10	4	0	-11 NO	number of trades left for this roster
rosteremp	SalarySum	int	10	4	0	0 NO	current salary total of starting lineup in roster
rosteremp	Wk	int	10	4	0	0 NO	current week of season for this roster
rosteremp	G1	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	G2	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	G3	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	G4	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	G5	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	F1	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	F2	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	F3	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	F4	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	F5	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	C1	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	C2	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	C3	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	SW	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	B1	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	B2	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	B3	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	B4	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	B5	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	B6	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	B7	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	B8	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	B9	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	B10	int	10	4	0 NULL	YES	BTG ID of player in this position for this week
rosteremp	B11	int	10	4	0 NULL	YES	BTG ID of player in this position for this week

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
rostertrans	transactionid	int	10	4	0 NULL	NO	Uniquely identifies a transaction
rostertrans	transactiontypeid	int	10	4	0 NULL	YES	ID for type of transaction
rostertrans	outplayer	int	10	4	0 NULL	YES	ID of player being moved out of a slot
rostertrans	inplayer	int	10	4	0 NULL	YES	ID of player being moved into a slot
rostertrans	date	datetime	23	16	3 NULL	YES	date and time of transaction
rostertrans	rosterid	int	10	4	0 NULL	YES	ID of roster transaction was performed on
rostertrans	status	bit	1	1	0 NULL	YES	Flag for whether transaction was completed



<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS; LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
scoringvalue	scoringvalueid	int	10	4	0 NULL	NO	Uniquely identifies a scoring value type
scoringvalue	type	varchar	50	50 NULL	NULL	NO	description of scoring type
scoringvalue	value	int	10	4	0 NULL	NO	Point value for scoring type

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS; LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
team	teamid	int	10	4	0 NULL	NO	Uniquely identifies a professional team
team	teamabbr	char	3	3 NULL	NULL	NO	Team's abbreviation
team	teamloc	varchar	30	30 NULL	NULL	YES	Location of professional team
team	teamname	varchar	30	30 NULL	NULL	YES	Name of team
team	teamconf	int	10	4	0 NULL	YES	Conference team is in
team	scoreid	int	10	4	0 NULL	YES	
team	statsincid	int	10	4	0 NULL	NO	Vendor's unique id for team
team	bdsportsid	int	10	4	0 NULL	YES	Clients unique id for team
team	ZipCode	varchar	50	50 NULL	NULL	YES	zipcode of team's location

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS; LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
transactiontype	TransactionTypeid	int	10	4	0 NULL	NO	Uniquely identifies a transaction type
transactiontype	Description	varchar	255	255 NULL	NULL	YES	Description of transaction type (add, drop, game purch)

(note: users table was not created by author of this project but is listed here as one of two tables referenced outside of the project's primary database  
...it is clearly not normalized)

<u>TABLE NAME</u>	<u>COLUMN NAME</u>	<u>TYPE NAME</u>	<u>PRECIS; LENGTH</u>	<u>SCALE</u>	<u>COLUMN DEF</u>	<u>IS NULLABLE</u>	<u>DESC</u>
Users	UserID	int	10	4	0 NULL	NO	Uniquely identifies a user
Users	FirstName	nvarchar	20	40 NULL	NULL	0 NO	user first name
Users	MiddleInitial	nvarchar	20	40 NULL	NULL	YES	user last name
Users	LastName	nvarchar	20	40 NULL	NULL	0 NO	
Users	Address1	nvarchar	60	120 NULL	NULL	0 NO	
Users	Address2	nvarchar	60	120 NULL	NULL	0 NO	
Users	City	nvarchar	20	40 NULL	NULL	0 NO	
Users	CountryID	int	10	4	0	0 NO	
Users	StateID	int	10	4	0	0 NO	
Users	ZipCode	nvarchar	5	10 NULL	NULL	0 NO	
Users	Phone	nvarchar	24	48 NULL	NULL	0 NO	
Users	Email	nvarchar	60	120 NULL	NULL	0 NO	
Users	HaveCCInfo	bit	1	1	0	0 YES	flag for whether we have credit card on file
Users	CCType	int	10	4	0	0 YES	credit card type
Users	CCName	nvarchar	50	100 NULL	NULL	0 YES	credit card name
Users	CCExpDateMonth	char	2	2 NULL	NULL	0 YES	credit card expiration month
Users	CCExpDateYear	char	2	2 NULL	NULL	0 YES	credit card expiration year
Users	CCNumber	nvarchar	20	40 NULL	NULL	0 YES	credit card number
Users	CCAddress	nvarchar	10	20 NULL	NULL	0 YES	credit card address
Users	CCCity	nvarchar	10	20 NULL	NULL	0 YES	credit card city

Users	CCStateID	int	10	4	0	0 YES	credit card stateid
Users	CCZip	int	10	4	0 NULL	YES	credit card zip
Users	CCCountryID	int	10	4	0 NULL	YES	credit card country id
Users	username	nvarchar	50	100 NULL	0 NO	0 NO	username
Users	password	nvarchar	50	100 NULL	0 NO	0 NO	password
Users	Question	nvarchar	50	100 NULL	0 NO	0 NO	question for if password is forgotten
Users	Answer	nvarchar	50	100 NULL	0 NO	0 NO	answer for if password is forgotten
Users	ReceiveMail	nvarchar	50	100 NULL	0 NO	0 NO	
Users	Purchased	bit	1	1	0	0 NO	
Users	private	bit	1	1	0	0 NO	questionaire info
Users	IncomeID	int	10	4	0	0 NO	questionaire info
Users	AgeID	int	10	4	0	0 NO	questionaire info
Users	GenderID	int	10	4	0	0 NO	questionaire info
Users	OccupationID	int	10	4	0	0 NO	questionaire info
Users	FavPubID	int	10	4	0	0 NO	questionaire info
Users	Updates	bit	1	1	0	0 NO	questionaire info
Users	AffiliateInfo	bit	1	1	0	0 NO	questionaire info
Users	HeardAbouID	int	10	4	0	0 NO	questionaire info
Users	CobrandID	int	10	4	0	-4 NO	cobrand which user is associated with
Users	lastlogin	datetime	23	16	3 NULL	YES	date user last logged into site
Users	creationdate	datetime	23	16	3 NULL	YES	user creation date
Users	updated	datetime	23	16	3 NULL	YES	last updated
Users	acct_bal	money	19	21	4	0 YES	acct balance for all games
Users	active	bit	1	1	0	-1 NO	whether they are an active user in the site

(note: users table was not created by author of this project but is listed here as one of two tables referenced outside of the project's primary database)

TABLE NAME	COLUMN NAME	TYPE NAME	PREC:	LENGTH	SCALE	COLUMN DEF	IS NULLABLE	DESC
Cobranded	CobrandID	int	10	4	0 NULL	NO	NO	Uniquely identifies a cobrand site
Cobranded	Cobrand	nvarchar	50	100 NULL	NULL	YES	YES	Cobrand site name
Cobranded	ServerName	varchar	50	50 NULL	NULL	YES	YES	Cobrand server
Cobranded	ServerName2	varchar	50	50 NULL	NULL	YES	YES	Cobrand server
Cobranded	ServerName3	varchar	50	50 NULL	NULL	YES	YES	Cobrand server
Cobranded	CobrandRoot	varchar	50	50 NULL	NULL	YES	YES	Cobrand Root
Cobranded	href	nvarchar	50	100 NULL	NULL	YES	YES	
Cobranded	stylesheet	nvarchar	50	100 NULL	NULL	YES	YES	
Cobranded	java	nvarchar	70	140 NULL	NULL	YES	YES	
Cobranded	logo	nvarchar	50	100 NULL	NULL	YES	YES	
Cobranded	topnav	nvarchar	50	100 NULL	NULL	YES	YES	
Cobranded	bitnav	nvarchar	50	100 NULL	NULL	YES	YES	
Cobranded	background	nvarchar	50	100 NULL	NULL	YES	YES	
Cobranded	bgscolor	nvarchar	50	100 NULL	NULL	YES	YES	
Cobranded	title	nvarchar	50	100 NULL	NULL	YES	YES	
Cobranded	http_host	varchar	50	50 NULL	NULL	YES	YES	
Cobranded	toprowtemplate	varchar	70	70 NULL	NULL	YES	YES	
Cobranded	root	varchar	50	50 NULL	NULL	YES	YES	
Cobranded	contentbanner	varchar	70	70 NULL	NULL	YES	YES	
Cobranded	commishEmail	varchar	70	70 NULL	NULL	YES	YES	

Cobranded	darknav	char	255	255 NULL	NULL	YES
Cobranded	lightback	char	255	255 NULL	NULL	YES
Cobranded	navheader	char	255	255 NULL	NULL	YES
Cobranded	navheaderType	char	255	255 NULL	NULL	YES
Cobranded	nava	char	255	255 NULL	NULL	YES
Cobranded	fontface	char	255	255 NULL	NULL	YES
Cobranded	contentheader	char	255	255 NULL	NULL	YES
Cobranded	contentheaderType	char	255	255 NULL	NULL	YES
Cobranded	darkbodyType	char	255	255 NULL	NULL	YES
Cobranded	lightbodyType	char	255	255 NULL	NULL	YES
Cobranded	footer	char	255	255 NULL	NULL	YES
Cobranded	clubhouse	char	255	255 NULL	NULL	YES
Cobranded	salaryover	char	255	255 NULL	NULL	YES
Cobranded	salaryunder	char	255	255 NULL	NULL	YES
Cobranded	rosterlight	char	255	255 NULL	NULL	YES
Cobranded	rosterdark	char	255	255 NULL	NULL	YES
Cobranded	rosterrow	char	255	255 NULL	NULL	YES
Cobranded	gamehighlight	char	255	255 NULL	NULL	YES
Cobranded	gamebkj	char	255	255 NULL	NULL	YES
Cobranded	gamefont	char	255	255 NULL	NULL	YES

## **APPENDIX E**

CREATE PROCEDURE ActivatePlayer

    @rid          int,  
    @toActive     int,  
    @benchSlot    char(3),  
    @actSlot      char(3),  
    @tablename    varchar(25),  
    @gameid      int

AS

DECLARE

    @Player\_in\_slot      int,  
    @newBenchID int,  
    @sql\_stmtnt      varchar(255),  
    @toBench          int

EXEC IdInSlot @actSlot,  
              @rid,  
              @player\_in\_slot OUTPUT,  
              @tablename

if @Player\_in\_slot is not null and @player\_in\_slot > 0  
    BEGIN

        set @newBenchID = @Player\_in\_slot  
    END

else

    BEGIN  
        set @newBenchID = 0  
    END

set @sql\_stmtnt = 'update '+@tablename+  
                  ' set '+@actSlot +'=' + ltrim(str(@toActive))+', '  
                  +@benchSlot+' = '+ltrim(str(@newBenchID))+  
                  ' where RosterID = '+ltrim(str(@rid))  
exec(@sql\_stmtnt)

set @sql\_stmtnt = 'insert rostertrans (TransactionTypeID, date, OutPlayer,  
  InPlayer, rosterid,status)  
                  VALUES (25,CURRENT\_TIMESTAMP,'+ltrim(str(@toActive))+', '  
                          +ltrim(rtrim(@newBenchID))+', '+ltrim(str(@rid))+',0)'  
exec(@sql\_stmtnt)

exec updateSalaryCap @rid, @gameid, 'Y'

CREATE PROCEDURE AddPlayer

    @slot        char(3),  
    @rid         int,  
    @pid         int,  
    @gameid      int

AS

DECLARE

    @player\_in\_slot        int,  
    @sql\_update\_string     varchar(255),  
    @sql\_trans\_string      varchar(255),  
    @livegame             int,  
    @currentbalance        money,  
    @charge\_amount         money,  
    @sql\_strmnt            varchar(255)

EXEC IdInSlot

    @slot,  
    @rid,  
    @player\_in\_slot OUTPUT,  
    'RosterTemp'

    BEGIN

        IF left(@slot,1) = 'B'

            BEGIN

                set @sql\_update\_string = 'update RosterTemp set  
                    '+rtrim(@slot)+' = ' +ltrim(str(@pid))+  
                    'where RosterID = '+ltrim(str(@rid))

            END

        ELSE

            BEGIN

                set @sql\_update\_string =  
                    'update RosterTemp set '+rtrim(@slot)+' = '  
                    '+ltrim(str(@pid))+ ' where RosterID =  
                    '+ltrim(str(@rid))

            END

                set @sql\_trans\_string =  
                    'insert rostertrans (TransactionTypeID, date,  
                    OutPlayer, InPlayer, rosterid,status)  
                    VALUES (21,CURRENT\_TIMESTAMP,  
                    0,'+ltrim(@pid)+' , ' +ltrim(@rid)+' ,0)'

                EXEC(@sql\_update\_string)

                EXEC(@sql\_trans\_string)

        /\* update salary cap info \*/

EXEC UpdateSalaryCap

    @rid,  
    @gameid,  
    'Y'

```

set @livegame = (select livegame from game where gameid = @gameid)

/* Check to see if game is live */
if @livegame = 1
    BEGIN
        update RosterTemp
            set trades = trades - 1
            where rosterid=@rid
        /* set @sql_stmt = 'set @currentbalance =
            (select accountbalance
            from RosterTemp)'

        exec(@sql_stmt)
        If left(@slot,2) = 'BE'
            BEGIN
                set @charge_amount = 2
            END
        Else
            BEGIN
                set @charge_amount = 5
            END
        If @currentbalance >= @charge_amount
            BEGIN
                set @sql_stmt =
                'update RosterTemp set accountbalance =
                @currentbalance - @charge_amount'

                exec(@sql_stmt)
            END*/

        /* Else Charge Credit Card */

    END
END
END

```

CREATE PROCEDURE BenchPlayer

```
@bench_slot    varchar(3),
@act_slot       varchar(3),
@rid            int,
@pid            int,
@toActivePID    int,
@tablename      varchar(255),
@gameid         int
```

AS

DECLARE

```
@player_in_slot    int,
@sql_act_update_string varchar(255),
@sql_bench_update_string varchar(255),
@sql_trans_string   varchar(255),
@playerid           int,
@offdefstatus       int
```

set @player\_in\_slot = 0

EXEC IdInSlot

```
@act_slot,
@rid,
@player_in_slot OUTPUT,
@tablename
```

If @player\_in\_slot <> 0

BEGIN

```
set @sql_act_update_string = 'update '+@tablename+
                             ' set '+@act_slot+' = '+ltrim(@toActivePID)+
                             ' where RosterID='+ltrim(@rid)
```

```
set @sql_bench_update_string = 'update '+@tablename+
                                ' set '+@bench_slot+' = '+ltrim(rtrim(@pid))+
                                ' where RosterID='+ltrim(@rid)
```

```
set @sql_trans_string = 'insert rostertrans (TransactionTypeID, date,
                                OutPlayer, InPlayer, rosterid,status)
                                VALUES(23,CURRENT_TIMESTAMP,'+
                                ltrim(@toActivePID)+' , '+ltrim(rtrim(@pid))+' ,'+
                                ltrim(str(@rid))+' ,0)'
```

```
EXEC(@sql_act_update_string)
EXEC(@sql_bench_update_string)
EXEC(@sql_trans_string)
```

```
EXEC updatesalarycap
    @rid,
    @gameid,
```



```
                'Y'  
            END  
Else  
    BEGIN  
        /* No player in field to drop/trade */  
        raiserror(50005,16,1)  
    END
```

CREATE PROCEDURE createRosterCopy

    @rid    int,  
    @del    char

AS

DECLARE

    @sql\_stmtnt    varchar(255),  
    @sql\_stmtnt2    varchar(255),  
    @sql\_stmtnt3    varchar(255),  
    @checkNum    int

BEGIN

    set @sql\_stmtnt = 'select RosterID  
                      from RosterTemp  
                      where RosterID= '+ltrim(str(@rid))  
    exec(@sql\_stmtnt)

    if @@rowcount = 0

        BEGIN

            set @sql\_stmtnt = 'insert into RosterTemp  
                              select \* from Roster  
                              where RosterID = '+ltrim(str(@rid))

        END

        exec(@sql\_stmtnt)

    if @del = 'Y'

        BEGIN

            set @sql\_stmtnt2 = 'delete from RosterTemp  
                              where RosterID='+ltrim(str(@rid))

            delete from rostertrans  
                      where rosterid = @rid and status = 0

            set @sql\_stmtnt3 = 'insert into RosterTemp  
                              select \*  
                              from Roster  
                              where RosterID = '+ltrim(str(@rid))

        END

        exec(@sql\_stmtnt2)  
        exec(@sql\_stmtnt3)

    delete from DrpPlayersTemp  
            where RosterID = @rid

END

CREATE PROCEDURE CreateTeam

```
@uid          int,
@teamname      varchar(255),
@gameid        int,
@cobrandid     int,
@reservations  int,
@rid           int output,
@lid           int output,
@rid_string    varchar(255) output
```

AS

DECLARE

```
@LeagueCount    int,
@requiredSpots   int,
@league_to_assign int,
@sql_stmt        varchar(255),
@counter         int,
@assignName      int,
@uid_in_league   int,
@thisRID         int,
@leagueid        int,
@leaguename      varchar(255)
```

```
if @reservations = 0
    BEGIN
        set @requiredSpots = 1
    END
else
    BEGIN
        set @requiredSpots = @reservations + 1
    END
```

/\* If no leagues exists than create one \*/

```
set @LeagueCount = (select count(leagueid) from league where cobrandid=@cobrandid
and gameid=@gameid)
```

```
If @LeagueCount = 0
```

```
/* Create League */
```

```
BEGIN
```

```
insert into league (gameid, commissionerID, availableSpots,
private, cobrandid)
```

```
values (@gameid, 0, 25, 0, @cobrandid)
```

```
END
```

```
set @leagueid=(Select max(leagueid) from league)
```

```
/* assign unique league name*/
```

```
set @leaguename= (select cobrandroot from fantasy.dbo.cobranded where
cobrandid=@cobrandid)
```

```
update league set leaguename=@leaguename+right(str(@leagueid),3) where
leagueid=@leagueid
```

```

set @league_to_assign = -1
create table #temp_not_these_leagues (league int)

WHILE @league_to_assign = -1
BEGIN
    set @league_to_assign = (select min(leagueid)
                             from league
                             where availableSpots > (select
                                                         @requiredSpots)
                             and leagueid NOT IN (select league
                                                  from #temp_not_these_leagues)
                                                  and cobrandid=@cobrandid
                                                  and gameid=@gameid)

    select UserID
    from Roster
    where LeagueID = (select @league_to_assign) and userid =
                    (select @uid)

    if @@rowcount > 0
    /*if @uid_in_league = @uid*/
    BEGIN
        set @sql_stmnt = 'insert into #temp_not_these_leagues
                          (league) values
                          ('+ltrim(str(@league_to_assign))+')'

        exec(@sql_stmnt)
        set @league_to_assign = -1
    CONTINUE
    END

    if @league_to_assign IS NULL
    BEGIN
        BEGIN TRANSACTION
        /* Create League */
        insert into league (gameid, commissionerID,
                           availableSpots, private, cobrandid)
        values (@gameid, 0, 25, 0, @cobrandid)
        set @league_to_assign = (select max(leagueid)
                                from league)

        COMMIT TRANSACTION
    END
END

drop table #temp_not_these_leagues

/* Create new team & assign to league*/
BEGIN TRANSACTION
    set @counter = 1
    set @assignName = 1
    set @thisRid = 0
    set @rid_string = ''

```

```

WHILE @counter <= @requiredSpots
    BEGIN
        IF @assignName = 1
            BEGIN
                set @sql_stmt = 'INSERT INTO Roster
                    (TeamName, LeagueID, UserID,
                    Trades, SalarySum)
                    VALUES +quotename(@teamname,'"') +
                    ', '+ltrim(str(@league_to_assign))+', '+
                    ltrim(str(@uid))+',11,0)'

                exec(@sql_stmt)
                set @sql_stmt = 'update League set
                    availableSpots = availableSpots - 1
                    where leagueid =
                    '+ltrim(str(@league_to_assign))

                exec (@sql_stmt)
                set @rid = (select max(rosterid)
                    from Roster)
                set @lid = @league_to_assign
                set @assignName = 0
                set @counter = @counter + 1
            CONTINUE
            END
        ELSE
            BEGIN
                set @sql_stmt = 'INSERT INTO Roster
                    (LeagueID, reservation)
                    VALUES
                    ('+ltrim(str(@league_to_assign))+',1)'

                exec(@sql_stmt)
                set @sql_stmt = 'update League
                    set availableSpots = availableSpots - 1
                    where leagueid =
                    '+ltrim(str(@league_to_assign))

                exec (@sql_stmt)
                set @thisRID = (select max(RosterID)
                    from Roster)
                set @rid_string = ltrim(str(@thisRID))
                    +', '+ltrim(rtrim(@rid_string))

                set @counter = @counter + 1
            CONTINUE
            END
        END
    END

COMMIT TRANSACTION

```

CREATE PROCEDURE DropPlayer1

```
@slot      char(3),
@rid        int,
@pid        int,
@gameid     int
```

AS

DECLARE

```
@player_in_slot      int,
@sql_update_string    varchar(255),
@sql_trans_string     varchar(255),
@sql_stmnt            varchar(255),
@status              bit
```

```
set @slot=ltrim(rtrim(@slot))
set @player_in_slot = 0
```

```
EXEC IdInSlot @slot,
              @rid,
              @player_in_slot OUTPUT,
              'RosterTemp'
```

```
print @slot
print @rid
print @pid
print 'player in slot'
print @player_in_slot
select @player_in_slot
```

```
if @player_in_slot <> 0
```

```
    BEGIN
```

```
        /* update roster */
```

```
        set @sql_update_string = 'update RosterTemp
                                   set '+'@slot+' = 0
                                   where RosterID='+ ltrim(str(@rid))
```

```
        /* update transaction log */
```

```
        set @sql_trans_string = 'insert rostertrans (TransactionTypeID, date,
                                   OutPlayer, InPlayer, rosterid,status)
                                   VALUES
                                   (22,CURRENT_TIMESTAMP, '+ltrim(str(@pid))+',
                                   0, '+ltrim(str(@rid))+',0)'
```

```
        EXEC(@sql_update_string)
```

```
        EXEC(@sql_trans_string)
```

```
        insert Drp (rosterid, btgid, dropdate, status)
                   values (@rid, @pid, getdate(), @status)
```

```
    END
```

```

CREATE PROCEDURE getActivateInfo
    @rid int,
    @pos char(3)
AS
declare
    @sql_string varchar(50),
    @sql_stmt varchar(255)

CREATE TABLE #current_roster (slot varchar(3), pid int)

insert #current_roster (slot, pid) EXEC GetSlotAndId @rid, 'Y'

select @sql_string =
    case @pos
        when 'G' then 'G1,G2,G3,G4,G5'
        when 'F' then 'F1,F2,F3,F4,F5'
        when 'C' then 'C1,C2,C3'
        when 'S' then 'G1,G2,G3,G4,G5,F1,F2,F3,F4,F5,C1,C2,C3'
    end

select cr.slot, cr.pid, po.positionAbbr, pl.fname, pl.lname, pl.salary, pl.salary30,
    pl.btgid, pl.bdsportsID, tm.teamid, tm.teamloc, tm.teamname, tm.teamabbr,
    po.positionAbbr
from #current_roster cr
    LEFT OUTER JOIN player pl ON (cr.pid = pl.btgid)
    LEFT OUTER JOIN position po ON (po.positionid = pl.positionid)
    LEFT OUTER JOIN Team tm ON (pl.TeamID=tm.TeamID)
    LEFT OUTER JOIN RosterPos pos ON (cr.slot = pos.rosterposid)
where left(cr.slot,1) = @pos
order by pos.slotOrder

drop table #current_roster

```

Temp

' + ltrim(str(@

(Transaction'  
er, rosterid,st

IMESTAMP,'+  
)+' ,0)'

atus)

CREATE PROCEDURE DropPlayer1

```
@slot      char(3),
@rid        int,
@pid        int,
@gameid     int
```

AS

DECLARE

```
@player_in_slot      int,
@sql_update_string    varchar(255),
@sql_trans_string     varchar(255),
@sql_stmnt            varchar(255),
@status              bit
```

```
set @slot=ltrim(rtrim(@slot))
set @player_in_slot = 0
```

```
EXEC IdInSlot @slot,
              @rid,
              @player_in_slot OUTPUT,
              'RosterTemp'
```

```
print @slot
print @rid
print @pid
print 'player in slot'
print @player_in_slot
select @player_in_slot
```

```
if @player_in_slot <> 0
```

```
    BEGIN
```

```
        /* update roster */
```

```
        set @sql_update_string = 'update RosterTemp
                                   set '+'@slot+' = 0
                                   where RosterID=' + ltrim(str(@rid))
```

```
        /* update transaction log */
```

```
        set @sql_trans_string = 'insert rostertrans (TransactionTypeID, date,
                                                       OutPlayer, InPlayer, rosterid,status)
                                   VALUES
                                   (22,CURRENT_TIMESTAMP, '+'ltrim(str(@pid))+'
                                   ,0, '+'ltrim(str(@rid))+' ,0)'
```

```
        EXEC(@sql_update_string)
```

```
        EXEC(@sql_trans_string)
```

```
        insert Drp (rosterid, btgid, dropdate, status)
                   values (@rid, @pid, getdate(), @status)
```

```
    END
```



CREATE PROCEDURE GetAddInfo

@rid int,  
@pos varchar(3),  
@rtype varchar(3)

AS

CREATE TABLE #current\_roster (slot varchar(3), pid int)

insert #current\_roster (slot, pid) EXEC GetSlotAndId  
@rid,  
'Y'

/\* Get Offensive Player List \*/

/\*UNLISTED PLAYERS\*/

select pl.\*, po.\*, t.teamname, t.teamloc, t.teamabbr  
from Player pl, position po, team t  
where pl.positionID=po.positionID  
AND po.positionAbbr=left(@pos,1)  
AND pl.teamid = t.teamid  
AND pl.listed = 0  
AND pl.salary > 0  
AND pl.btgid not in (select pid from #current\_roster)  
order by pl.bdsportsid

/\*LISTED PLAYERS\*/

select pl.\*, po.\*, t.teamname, t.teamloc, t.teamabbr  
from player pl, position po, team t  
where pl.positionID=po.positionID  
AND po.positionAbbr=left(@pos,1)  
AND pl.teamid = t.teamid  
AND pl.listed = 1  
AND pl.btgid not in (select pid from #current\_roster)  
order by pl.bdsportsid

CREATE PROCEDURE getBenchInfo

@rid int,  
@pos char(3)

AS

DECLARE

@sql\_string varchar(50),  
@sql\_stmnt varchar(255)

CREATE TABLE #current\_roster (slot varchar(3), pid int)  
insert #current\_roster (slot, pid) EXEC GetSlotAndId @rid, 'Y'

BEGIN

if (left(@pos,1) <> 'S')

```

BEGIN
    select cr.slot, cr.pid, po.positionAbbr, pl.fname, pl.lname, pl.salary,
           pl.salary30, pl.btgid, pl.bdsportsid, tm.teamid, tm.teamloc,
           tm.teamname, tm.teamabbr, po.positionAbbr
    from #current_roster cr
    LEFT OUTER JOIN player pl ON (cr.pid = pl.btgid)
    LEFT OUTER JOIN position po ON (po.positionid = pl.positionid)
    LEFT OUTER JOIN Team tm ON (pl.TeamID=tm.TeamID)
    left outer join RosterPos pos ON (cr.slot = pos.rosterposID)
    where left(cr.slot,1) = 'B'
    and po.positionAbbr = @pos
    order by pos.slotOrder
END
else
    BEGIN
        select cr.slot, cr.pid, po.positionAbbr, pl.fname, pl.lname, pl.salary,
               pl.salary30, pl.btgid, pl.bdsportsid, tm.teamid, tm.teamloc,
               tm.teamname, tm.teamabbr, po.positionAbbr
        from #current_roster cr
        LEFT OUTER JOIN player pl ON (cr.pid = pl.btgid)
        LEFT OUTER JOIN position po ON (po.positionid = pl.positionid)
        LEFT OUTER JOIN Team tm ON (pl.TeamID=tm.TeamID)
        left outer join RosterPos pos ON (cr.slot = pos.rosterposID)
        where left(cr.slot,1) = 'B'
        order by pos.slotOrder
    END
END
drop table #current_roster

```

CREATE PROCEDURE GetCurrentRoster

@rid int,  
@update char,  
@gameid int,  
@inj\_num int OUTPUT,  
@drp\_num int OUTPUT

AS

DECLARE

@this\_slot char(3),  
@my\_int int,  
@sql varchar(1000),  
@tablename varchar(1000),  
@this\_bench\_slot char(3),  
@bench\_status\_slot char(10),  
@sql\_stmt varchar(1000),  
@player\_in\_slot int,  
@name varchar(50),  
@Bname varchar(50),  
@salary int,  
@Bsalary int,  
@bdsportsID int,  
@BbdsportsID int,  
@internalID int,  
@BinternalID int

if @update = 'Y'  
set @tablename = 'RosterTemp'  
else  
set @tablename = 'Roster'

/\* These statements use the GetSlotAndId proc to produce #current\_roster table  
(which holds the slot and playerid) \*/

CREATE TABLE #current\_roster (slot varchar(3), pid int)  
insert #current\_roster (slot, pid) EXEC GetSlotAndId  
@rid,  
@update

CREATE TABLE #current\_active (slot char(3), BDSportsID int, btgID int, outname  
varchar(50), salary int, pos char(3), tm char(3), Minutes int, Pts int, FGM int, FGA int,  
FTM int, FTA int, [3GM] int, [3GA] int, OReb int, DReb int, Reb int, Ast int,  
St int, Blk int, TurnO int, PFoul int, TFoul int)

select \* from #current\_roster

DECLARE active\_cursor CURSOR FORWARD\_ONLY LOCAL for  
select slot from #current\_roster where left(slot,1)<>'B'

OPEN active\_cursor  
FETCH active\_cursor into @this\_slot  
DECLARE @minutes int,  
@pts int,

```

@FGM int,
@FGA int,
@FTM int,
@FTA int,
@3GM int,
@3GA int,
@OReb int,
@DReb int,
@Reb int,
@Ast int,
@St int,
@Blk int,
@TurnO int,
@PFoul int,
@TFoul int,
@pos char(3),
@tm char(3)

```

```

WHILE (@@FETCH_STATUS=0)
  BEGIN

```

```

    set @sql_stmt = 'select '+@this_slot+' from '+@tablename+
                    ' where RosterID= '+ltrim(str(@rid))

```

```

    exec(@sql_stmt)

```

```

    if @@rowcount > 0

```

```

        BEGIN

```

```

            EXEC IdInSlot @this_slot, @rid,
                        @player_in_slot OUTPUT, @tablename

```

```

            set @name = (select fname+' '+lname from player
                        where btgid = (select @player_in_slot))

```

```

            set @internalID = (select btgid from player
                        where btgid = (select @player_in_slot))

```

```

            set @minutes = (select sum(ips.[min]) from
                        importplayerstats ips,
                        player pl
                        where pl.btgid =(select @player_in_slot)
                        and pl.statsincid = ips.ID#)

```

```

            set @pts = (select sum( ips.pts)
                        from importplayerstats ips, player pl
                        where pl.btgid =(select @player_in_slot)
                        and pl.statsincid = ips.ID#)

```

```

            set @FGM = (select sum(ips.FGM)
                        from importplayerstats ips,
                        player pl
                        where pl.btgid =(select @player_in_slot)
                        and pl.statsincid = ips.ID#)

```

```

            set @FGA = (select sum(ips.FGA)
                        from importplayerstats ips,
                        player pl
                        where pl.btgid =(select @player_in_slot)
                        and pl.statsincid = ips.ID#)

```

```

            set @FTM = (select sum(ips.FTM)
                        from importplayerstats ips,
                        player pl
                        where pl.btgid =(select @player_in_slot)
                        and pl.statsincid = ips.ID#)

```

```

set @FTA = (select sum(ips.FTA)
            from importplayerstats ips,
            player pl
            where pl.btgid =(select @player_in_slot)
            and pl.statsincid = ips.ID#)
set @3GM = (select sum(ips.[3GM])
            from importplayerstats ips,
            player pl
            where pl.btgid =(select @player_in_slot)
            and pl.statsincid = ips.ID#)
set @3GA = (select sum(ips.[3GA])
            from importplayerstats ips,
            player pl
            where pl.btgid =(select @player_in_slot)
            and pl.statsincid = ips.ID#)
set @OReb = (select sum(ips.OReb)
            from importplayerstats ips,
            player pl
            where pl.btgid =(select @player_in_slot)
            and pl.statsincid = ips.ID#)
set @DReb = (select sum(ips.DReb)
            from importplayerstats ips,
            player pl
            where pl.btgid =(select @player_in_slot)
            and pl.statsincid = ips.ID#)

set @Reb = (select sum(ips.Reb)
            from importplayerstats ips,
            player pl
            where pl.btgid =(select @player_in_slot)
            and pl.statsincid = ips.ID#)
set @Ast = (select sum(ips.Ast)
            from importplayerstats ips,
            player pl
            where pl.btgid =(select @player_in_slot)
            and pl.statsincid = ips.ID#)
set @St = (select sum(ips.St)
            from importplayerstats ips,
            player pl
            where pl.btgid =(select @player_in_slot)
            and pl.statsincid = ips.ID#)
set @Blk = (select sum(ips.Blk)
            from importplayerstats ips,
            player pl
            where pl.btgid =(select @player_in_slot)
            and pl.statsincid = ips.ID#)
set @TurnO = (select sum(ips.[TO])
            from importplayerstats ips,
            player pl
            where pl.btgid =(select @player_in_slot)
            and pl.statsincid = ips.ID#)
set @PFoul = (select sum(ips.PF)
            from importplayerstats ips,
            player pl
            where pl.btgid =(select @player_in_slot)
            and pl.statsincid = ips.ID#)

```

```

set @TFoul = (select sum(ips.Tech)
              from importplayerstats ips,
              player pl
              where pl.btgid =(select @player_in_slot)
              and pl.statsincid = ips.ID#)
set @pos = (select distinct po.positionAbbr
            from position po,
            player pl
            where po.positionID = pl.positionID
            and pl.btgid = (select @player_in_slot))
set @tm = (select distinct tm.teamAbbr
           from team tm, player pl
           where pl.TeamID = tm.teamID
           and pl.btgid =(select @player_in_slot))
set @salary = (select Salary
               from player
               where btgid =(select @player_in_slot))
set @bdsportsID = (select bdsportsid
                  from player
                  where btgid =(select @player_in_slot))

insert #current_active (slot ,BDSportsID, btgID,
                       outname, salary, pos, tm, Minutes, Pts, FGM,
                       FGA , FTM, FTA, [3GM], [3GA], OReb, DReb,
                       Reb, Ast, St,Blk, TurnO, PFoul, TFoul)
values (@this_slot, @bdsportsID, @internalID,
        @name, @salary, @pos, @tm, @minutes,
        @pts,@FGM,@FGA,@FTM,@FTA,
        @3GM,@3GA,@OReb,@DReb,@Reb,@Ast,
        @St,@Blk, @TurnO,@PFoul,@TFoul)

END

        FETCH active_cursor into @this_slot
END

CLOSE active_cursor
DEALLOCATE active_cursor

CREATE TABLE #current_bench (slot char(3), thisID int, internalID int,
                             outname varchar(50), salary int, pos char(3), tm char(3),
                             Minutes int, Pts int, FGM int, FGA int, FTM int, FTA int,
                             [3GM] int, [3GA] int, OReb int, DReb int, Reb int, Ast int,
                             St int,Blk int, TurnO int, PFoul int, TFoul int)

select * from #current_roster

DECLARE bench_cursor CURSOR FORWARD_ONLY LOCAL for
                    select slot from #current_roster where left(slot,1)='B'

OPEN bench_cursor
FETCH bench_cursor into @this_bench_slot
DECLARE      @Bminutes int,
             @Bpts int,
             @BFGM int,
             @BFGA int,
             @BFTM int,
             @BFTA int,
             @B3GM int,

```

```

    @B3GA int,
    @BOReb int,
    @BDReb int,
    @BReb int,
    @BAst int,
    @BSt int,
    @BBlk int,
    @BTurnO int,
    @BPFoul int,
    @BTFoul int,
    @Bpos char(3),
    @Btm char(3)

```

```

WHILE (@@FETCH_STATUS=0)

```

```

    BEGIN

```

```

        set @sql_stmt='select '+@this_bench_slot+' from '+@tablename+
        ' where RosterID= '+ltrim(str(@rid))

```

```

        exec(@sql_stmt)

```

```

        if @@rowcount > 0

```

```

            BEGIN

```

```

                EXEC IdInSlot @this_bench_slot, @rid,
                @player_in_slot OUTPUT, @tablename

```

```

                set @Bname = (select fname+' '+lname
                from player where btgid =
                (select @player_in_slot))

```

```

                set @BinternalID=(select btgid from player where btgid=
                (select @player_in_slot))

```

```

                set @Bminutes = (select sum (ips.[min])
                from importplayerstats ips,
                player pl
                where pl.btgid =(select @player_in_slot)
                and pl.statsincid= ips.ID#)

```

```

                set @Bpts = (select sum( ips.pts)
                from importplayerstats ips,
                player pl
                where pl.btgid =(select @player_in_slot)
                and pl.statsincid = ips.ID#)

```

```

                set @BFGM = (select sum(ips.FGM)
                from importplayerstats ips,
                player pl
                where pl.btgid =(select @player_in_slot)
                and pl.statsincid = ips.ID#)

```

```

                set @BFGA = (select sum(ips.FGA)
                from importplayerstats ips,
                player pl
                where pl.btgid =(select @player_in_slot)
                and pl.statsincid = ips.ID#)

```

```

                set @BFTM = (select sum(ips.FTM)
                from importplayerstats ips,
                player pl
                where pl.btgid =(select @player_in_slot)
                and pl.statsincid = ips.ID#)

```

```

                set @BFTA = (select sum(ips.FTA)
                from importplayerstats ips,
                player pl
                where pl.btgid =(select @player_in_slot)

```

```

        and pl.statsincid = ips.ID#)
set @B3GM = (select sum(ips.[3GM])
             from importplayerstats ips,
             player pl
             where pl.btgid =(select @player_in_slot)
             and pl.statsincid = ips.ID#)
set @B3GA = (select sum(ips.[3GA])
             from importplayerstats ips,
             player pl
             where pl.btgid =(select @player_in_slot)
             and pl.statsincid = ips.ID#)
set @BOReb = (select sum(ips.OReb)
             from importplayerstats ips,
             player pl
             where pl.btgid =(select @player_in_slot)
             and pl.statsincid = ips.ID#)
set @BDReb = (select sum(ips.DReb)
             from importplayerstats ips,
             player pl
             where pl.btgid =(select @player_in_slot)
             and pl.statsincid = ips.ID#)
set @BReb = (select sum(ips.Reb)
             from importplayerstats ips,
             player pl
             where pl.btgid =(select @player_in_slot)
             and pl.statsincid = ips.ID#)
set @BAst = (select sum(ips.Ast)
             from importplayerstats ips,
             player pl
             where pl.btgid =(select @player_in_slot)
             and pl.statsincid = ips.ID#)
set @BSt = (select sum(ips.St)
             from importplayerstats ips,
             player pl
             where pl.btgid =(select @player_in_slot)
             and pl.statsincid = ips.ID#)
set @BBlk = (select sum(ips.Blk)
             from importplayerstats ips,
             player pl
             where pl.btgid =(select @player_in_slot)
             and pl.statsincid = ips.ID#)
set @BTurnO = (select sum(ips.[TO])
             from importplayerstats ips,
             player pl
             where pl.btgid =(select @player_in_slot)
             and pl.statsincid = ips.ID#)
set @BPFoul = (select sum(ips.PF)
             from importplayerstats ips,
             player pl
             where pl.btgid =(select @player_in_slot)
             and pl.statsincid = ips.ID#)
set @BTFoul = (select sum(ips.Tech)
             from importplayerstats ips,
             player pl
             where pl.btgid =(select @player_in_slot)
             and pl.statsincid = ips.ID#)

```



```

set @Bpos = (select distinct po.positionAbbr
             from position po,
             player pl
             where po.positionID = pl.positionID
             and pl.btgid = (select @player_in_slot))
set @Btm = (select distinct tm.teamAbbr
            from team tm, player pl
            where pl.TeamID = tm.teamID
            and pl.btgid = (select @player_in_slot))
set @Bsalary = (select Salary
                from player
                where btgid =(select @player_in_slot))
set @BbdsportsID = (select bdsportsid
                    from player
                    where btgid =(select @player_in_slot))

insert #current_bench (slot , thisID, internalID, outname,
                      salary, pos, tm, Minutes, Pts, FGM, FGA , FTM,
                      FTA, [3GM], [3GA], OReb, DReb, Reb, Ast,
                      St,Blk, TurnO, PFoul, TFoul)
values (@this_bench_slot, @BbdsportsID,
        @BinternalID, @Bname,@Bsalary, @Bpos,
        @Btm, @Bminutes,@Bpts,@BFGM,@BFGA,
        @BFTM,@BFTA,@B3GM,@B3GA,@BOREb,
        @BDRReb,@BReb,@BAst,@BSt,@BBlk,
        @BTurnO,@BPfoul,@BTFOul)

END

FETCH bench_cursor into @this_bench_slot
END

CLOSE bench_cursor
DEALLOCATE bench_cursor

/* Select Statement 1 -- Select Current Active Players */
select slot, bdsportsID, btgid, outname, salary, pos, tm, Minutes, Pts, FGM, FGA, FTM,
       FTA, [3GM] as gm3, [3GA], OReb, DReb, Reb, Ast, St, Blk, TurnO, PFoul, TFoul
from #current_active

/* Select Statement 2 -- Select Currently Benched Players */
select slot, thisID, internalID, outname, salary, pos, tm, Minutes, Pts, FGM, FGA, FTM,
       FTA, [3GM] as gm3, [3GA], OReb, DReb, Reb, Ast, St, Blk, TurnO, PFoul, TFoul
from #current_bench

/* Select Statement 4 -- select number of injured players currently in roster */
set @inj_num = (select count(cr.pid) from #current_roster cr where cr.pid in
               (select ID# from ImportInjury))
if @inj_num is null
    set @inj_num=0

/* Select Statement 5 -- select number of dropped players */
if @update = 'Y'

```

```

        set @drp_num = (select count(pl.btgid)
                        from dropplayer drp, player pl
                        where pl.btgid = drp.btgid
                        AND drp.rosterid=@rid)
else
        set @drp_num = (select count(pl.btgid)
                        from drp, player pl
                        where pl.btgid = drp.btgid
                        AND drp.rosterid=@rid)
if @drp_num is null
    set @drp_num=0

/* Select Statement 6 -List of players currenty in roster that are injured- result set 30 */
Select i.fname, i.lname, i.tm, p.pos, i.date, i.description
    from importinjury i, player p, #current_roster cr
    where p.statsincid=i.id#
    and p.btgid=cr.pid
    order by date desc

drop table #current_roster
drop table #current_bench

```

CREATE PROCEDURE GetLeagueStanding

    @rid        int,  
    @leagueid   int,  
    @standing   int output

AS

DECLARE

    @rid\_total   decimal(18,2)

Create table #TotalScores

    (rosterid int,  
    leagueid int,  
    Total decimal(18, 2))

insert #TotalScores (rosterid, leagueid, Total)

    Select rs.RosterID, r.leagueid, sum(rs.wktotal) as Total  
    from rosterscore\_arc rs, roster r  
    where rs.rosterid=r.rosterid  
    and r.leagueid=@leagueid  
    group by rs.rosterid, r.leagueid  
    order by Total desc

Select @rid\_total = (select sum(wktotal)  
                      from rosterscore\_arc  
                      where rosterid=@rid)

Select \*  
from #TotalScores  
where Total > @rid\_total

set @standing=@@rowcount + 1

drop table #TotalScores

CREATE PROCEDURE GetOverallStanding

    @rid        int,  
    @standing    int output

AS

DECLARE

    @rid\_total    decimal(18,2)

    Create table #TotalScores

        (rosterid int,  
        Total    decimal(18, 2))

    insert #TotalScores (rosterid, Total)

        Select rs.RosterID, sum(wktotal) as Total  
        from  rosterscore\_arc rs, roster r, league l  
        where  rs.rosterid=r.rosterid and  
              r.leagueid=l.leagueid and  
              l.internal = 0  
        group by rs.rosterid  
        order by Total desc

    Select @rid\_total = (select sum(wktotal)  
                          from rosterscore\_arc  
                          where rosterid=@rid)

    Select \*  
    from #TotalScores  
    where Total > @rid\_total

    set @standing=@@rowcount + 1

    Drop Table #TotalScores

CREATE PROCEDURE GetPosAndID

    @rid        int,  
    @update     char

AS

DECLARE

    @this\_pos    char(3),  
    @my\_int     int,  
    @sql         varchar(255),  
    @tablename   varchar(25),  
    @this\_pos\_status varchar(20)

BEGIN TRANSACTION

if upper(@update)='Y'

    set @tablename = 'RosterTemp'

else

    set @tablename = 'Roster'

CREATE TABLE #current\_roster (pos varchar(3), pid int)

DECLARE pos\_cursor CURSOR FORWARD\_ONLY LOCAL FOR  
    select rosterposID from RosterPos

OPEN pos\_cursor

FETCH pos\_cursor into @this\_pos

/\* This loop builds the table of slots and playerids \*/

WHILE (@@FETCH\_STATUS=0)

    BEGIN

        if left(@this\_pos,1)='B'

            BEGIN

                set @sql = 'insert #current\_roster (pos, pid)  
                            select '+quotename(@this\_pos,'')+'  
                                AS pos, '+@this\_pos+  
                            ' FROM '+@tablename+  
                            ' where RosterID='+ltrim(str(@rid))

            END

        else

            BEGIN

                set @sql = 'insert #current\_roster (pos, pid)  
                            select '+quotename(@this\_pos,'')+  
                                ' AS pos, '+@this\_pos+  
                            ' FROM '+@tablename+  
                            ' where RosterID='+ltrim(str(@rid))

            END

            EXEC(@sql)

            FETCH pos\_cursor into @this\_pos

    END

CLOSE pos\_cursor

DEALLOCATE pos\_cursor

select \* from #current\_roster

drop table #current\_roster

COMMIT TRANSACTION

CREATE PROCEDURE GetSlotAndID

    @rid        int,  
    @update     char

AS  
DECLARE

    @this\_slot  char(3),  
    @my\_int     int,  
    @sql        varchar(255),  
    @tablename  varchar(25)

BEGIN TRANSACTION  
    if upper(@update)='Y'  
        set @tablename = 'RosterTemp'  
    else  
        set @tablename = 'Roster'

CREATE TABLE #current\_roster (slot varchar(3), pid int)

DECLARE slot\_cursor CURSOR FORWARD\_ONLY LOCAL FOR  
                    select rosterposid from RosterPos

OPEN slot\_cursor  
FETCH slot\_cursor into @this\_slot

/\* This loop builds the table of slots and playerids \*/

WHILE (@@FETCH\_STATUS=0)  
BEGIN  
    /\* if left(@this\_slot,1)='B'  
    BEGIN  
        set @sql = 'insert #current\_roster (slot, pid)  
                    select '+quotename(@this\_slot,'')+  
                    ' AS slot, '+@this\_slot+  
                    ' FROM '+@tablename+  
                    ' where RosterID='+ltrim(str(@rid))  
    END  
    else  
        BEGIN\*/  
        set @sql = 'insert #current\_roster (slot, pid)  
                    select '+quotename(@this\_slot,'')+  
                    ' AS slot, '+@this\_slot+  
                    ' FROM '+@tablename+  
                    where RosterID='+ltrim(str(@rid))  
        /\*END\*/  
        EXEC (@sql)  
        FETCH slot\_cursor into @this\_slot  
    END  
CLOSE slot\_cursor  
DEALLOCATE slot\_cursor  
  
select \* from #current\_roster  
drop table #current\_roster

```

CREATE PROCEDURE IDInSlot
    @slot          varchar(3),
    @rosterid      int,
    @pid           int output,
    @tablename     varchar(50)

```

AS

BEGIN

DECLARE

```

    @sql_select_string  varchar(250),
    @sql_set_string     varchar(255)

```

create table #temp\_move\_table(id int)

```

set @sql_select_string = 'insert #temp_move_table select '+rtrim(@slot)+'
                        ' from '+@tablename+
                        ' where rosterid='+ltrim(str(@rosterid))

```

EXEC(@sql\_select\_string)

set @pid = (select id from #temp\_move\_table)

drop table #temp\_move\_table

END

```

CREATE PROCEDURE InjuryList

```

```

    @order          varchar(50)

```

AS

Declare

```

    @sqlStmt      varchar(1000)

```

BEGIN

if @order='fname'

set @order = 'i.fname'

else if @order='lname'

set @order='i.lname'

else if @order = 'tm'

set @order = 'i.tm'

else if @order='pos'

set @order='p.pos'

else if @order='description'

set @order='description'

else

set @order='i.date'

```

set @sqlStmt= 'Select i.fname, i.lname, i.tm, p.pos, i.date, i.description
from importinjury i, player p
where p.statsincid=i.id#
order by '+quotename(@order, '"')

```

exec @sqlStmt

END

/\* This procedure will move the swing player to a starting position in the lineup associated with that players position (G,F,C).  
It will move the person in that starting position to the bench and move a bench player to the swing position

Input: @rid Roster id (integer),  
@StOutSlot The slot in the starting lineup that starter to be moved currently occupies (char(3)),  
@SwInSlot Slot of bench player to be moved to Swing Pos (char(3)),  
@Action to update roster or not

Output: Udated Roster with swing in starting, starter benched, and bench activated to swing  
\*/

CREATE PROCEDURE SwingPlayer

@rid int, /\*roster id\*/  
@StOutSlot varchar(3), /\*slot of player to be benched\*/  
@SwInSlot varchar(3), /\*slot of bench player to be moved to swing pos\*/  
@Action varchar(1),  
@gameid int

AS

DECLARE

@SwOutId int, /\* id of player in swing to be moved inside active roster\*/  
@StOutId int, /\*id of player to be benched\*/  
@SwInId int, /\*id of bench player to be moved to swing pos\*/  
@tablename varchar(50),  
@SqlInsertString varchar(255),  
@SqlUpdateString varchar(255),  
@SqlUpdateString2 varchar(255),  
@SqlUpdateString3 varchar(255),  
@SqlTranString varchar(255)

BEGIN

Create table #Current\_Roster(slot char(3), pid int)  
insert #current\_roster EXEC GetslotAndID @rid, 'Y'

Set @SwOutId = (Select pid from #Current\_Roster where slot = 'SW')  
Set @StOutId = (Select pid from #Current\_Roster where slot = @StOutSlot)  
Set @SwInId = (Select pid from #Current\_Roster where slot = @SwInSlot)

if @StOutId is null  
set @stoutid=0  
if @SwInId is null  
set @SwInId=0

if upper(@action)='Y'  
set @tablename = 'RosterTemp'  
else  
set @tablename = 'Roster'



```

select * from rosteremp where rosterid=@rid

if (@@rowcount=0)
    BEGIN
        set @SqlInsertString = 'insert rosteremp
                                select * from roster
                                where RosterID = '+ltrim(str(@rid))

        exec (@SqlInsertString)
    END

select * from #current_roster
select * from #current_roster where slot='sw'
select * from #current_roster where slot='b4'
select @swinslot as swingin
select @stoutslot as starterout
select @SwOutId as swingoutid
select @StOutId as starteroutid
select @SwInId as swinginid
select @tablename

set @SqlUpdateString = 'update '+@tablename+
                        ' set '+@StOutSlot+' = '+quotename(@SwOutId,'')+
                        ' where RosterID = '+str(@rid)
set @SqlUpdateString2 = 'update '+@tablename+
                        ' set '+@SwInSlot+' = '+quotename(@StOutId,'')+
                        ' where RosterID = '+str(@rid)
set @SqlUpdateString3 = 'update '+@tablename+
                        ' set SW='+quotename(@SwInId,'')+
                        ' where RosterID = '+str(@rid)

/*Log transaction RosterTrans table with uncommitted status of 0*/
set @sqltranstring = 'insert rostertrans (TransactionTypeID, date, OutPlayer, InPlayer,
                                rosterid,status)
                        VALUES (26, CURRENT_TIMESTAMP, '+ltrim(str(@StOutId))+', '
                                +ltrim(str(@SwInId))+', '+ltrim(str(@rid))+', 0)'

exec (@SqlUpdateString)
exec (@SqlUpdateString2)
exec (@SqlUpdateString3)
exec (@sqltranstring)
exec UpdateSalaryCap @rid, @gameid, 'Y'
select @tablename

END

drop table #current_roster

```

/\* This proc aggregates data from the RosterScores and RosterStats tables -- creating the total points and stats for the roster \*/

CREATE PROCEDURE UpdateRosterScores

    @wk int

AS

    IF (CURSOR\_STATUS('local', 'RosterIDCursor') <> -3)

        BEGIN

            close RosterIDCursor

            deallocate RosterIDCursor

        END

DECLARE

    @rid int,

    @total\_score int

DECLARE RosterIDCursor CURSOR FAST\_FORWARD for

    select RosterID from Roster\_previousweek

Open RosterIDCursor

Fetch RosterIDCursor into @rid

while (@@fetch\_status = 0)

    BEGIN

        CREATE TABLE #current\_roster (slot varchar(3), pid int)

        insert #current\_roster (slot, pid) EXEC GetSlotAndId\_Previous

                                @rid, @wk

        insert RosterStat (rosterid, wk, fg, ft, fg3, rebounds, assists,  
                            steals, blocks)

        select @rid, @wk, sum(fg), sum(ft), sum(fg3), sum(rebounds),  
                sum(assists), sum(steals), sum(blocks)

        from playerstat

        where statsincid in (select statsincid  
                            from player pl, #current\_roster cr  
                            where cr.pid = pl.btgid  
                            and upper(left(cr.slot,1)) <> 'B')

        insert RosterScore (rosterid, wk, fg, ft, fg3, rebounds, assists,  
                            steals, blocks)

        select @rid, @wk, sum(fg), sum(ft), sum(fg3), sum(rebounds),  
                sum(assists), sum(steals), sum(blocks)

        from playerscore

        where statsincid in (select statsincid  
                            from player pl, #current\_roster cr  
                            where cr.pid = pl.btgid  
                            and upper(left(cr.slot,1)) <> 'B')

        set @total\_score = (select fg + ft + fg3 + rebounds + assists +  
                            steals + blocks

        from rosterscore

        where RosterID = @rid

        AND wk=@wk)

        if @total\_score IS NULL set @total\_score=0

        update rosterscore set wkTotal = @total\_score where RosterID= @rid

```
fetch RosterIDCursor into @rid  
drop table #current_roster  
END
```

```
close RosterIDCursor  
deallocate RosterIDCursor
```

```
CREATE PROCEDURE UpdateSalaryCap
```

```
    @rid          int,  
    @gameid       int,  
    @update       char
```

```
AS
```

```
declare
```

```
    @sql_stmtnt   varchar(255),  
    @tablename    varchar(255),  
    @b_var        char,  
    @offsalary    int,  
    @totalSalary  int
```

```
    set @b_var = 'B'
```

```
BEGIN TRANSACTION
```

```
CREATE TABLE #temp_roster (slot varchar(3), pid int)  
insert #temp_roster (slot, pid) EXEC GetSlotAndId @rid,  
                                                    @update
```

```
IF @update = 'Y'
```

```
    BEGIN
```

```
        set @tablename = 'RosterTemp'
```

```
    END
```

```
ELSE
```

```
    BEGIN
```

```
        set @tablename = 'Roster'
```

```
    END
```

```
if @gameid=102
```

```
    set @offSalary = (select sum(salary30)  
                      from player  
                      where btgid in (select pid  
                                       from #temp_roster  
                                       where upper(left(slot,1)) <> 'B' and pid <> 0))
```

```
else
```

```
    set @offSalary = (select sum(salary)  
                      from player  
                      where btgid in (select pid  
                                       from #temp_roster  
                                       where upper(left(slot,1)) <> 'B'  
                                       and pid <> 0))
```

```
if @offSalary is null
```

```
    BEGIN
```

```
        set @offsalary = 0
```

```
    END
```

```
set @totalSalary = @offSalary
```

```
/*select @offsalary as offense, @defsalary as defense, @totalSalary as total*/
```

```
set @sql_stmtnt = 'UPDATE '+@tablename+
```

```
' set salarysum = '+ltrim(str(@totalSalary))+  
' where RosterID = '+ltrim(rtrim(@rid))
```

```
exec (@sql_stmt)  
drop table #temp_roster
```

COMMIT TRANSACTION

```
CREATE PROCEDURE UpdateScores
```

```
    @wk    int
```

```
AS
```

```
Declare
```

```
    @id      int,  
    @fg      int,  
    @ft      int,  
    @3fg     int,  
    @stl     int,  
    @ast     int,  
    @blk     int,  
    @reb     int,
```

```
    @fgValue  int,  
    @ftValue  int,  
    @3fgValue int,  
    @rebValue int,  
    @astValue int,  
    @stlValue int,  
    @blkValue int,
```

```
    @fgScore  int,  
    @ftScore  int,  
    @3fgScore int,  
    @rebScore int,  
    @astScore int,  
    @stlScore int,  
    @blkScore int
```

```
Declare scoringcursor cursor for select id#, sum(fgm), sum([3gm]), sum(ftm), sum(reb), sum(ast),  
sum(st), sum(blk)
```

```
    from importplayerstats  
    group by id#
```

```
Open scoringcursor
```

```
Fetch scoringcursor into @id, @fg, @3fg, @ft, @reb, @ast, @stl, @blk
```

```
set @fgValue= (select value from ScoringValue where scoringvalueid=1)  
set @ftValue= (select value from ScoringValue where scoringvalueid=2)  
set @3fgValue= (select value from ScoringValue where scoringvalueid=3)  
set @rebValue = (select value from ScoringValue where scoringvalueid=4)  
set @astValue = (select value from ScoringValue where scoringvalueid=5)  
set @stlValue = (select value from ScoringValue where scoringvalueid=6)  
set @blkValue = (select value from ScoringValue where scoringvalueid=7)
```

```
While (@@fetch_status=0)
```

```
    BEGIN
```

```
        set @fgScore = @fg * @fgValue  
        set @ftScore = @ft * @ftValue
```

```
set @3fgScore = @3fg * @3fgValue
set @rebScore = @reb * @rebValue
set @astScore = @ast * @astValue
set @stlScore = @stl * @stlValue
set @blkScore = @blk * @blkValue
```

```
insert playerscore (statsincid, wk, fg, ft, fg3, rebounds,assists,
                    steals, blocks)
values (@id, @wk, @fgScore, @ftScore, @3fgScore,
        @rebScore, @astScore, @stlScore, @blkScore)
```

```
insert playerstat (statsincid, wk, fg, ft, fg3, rebounds,assists,
                   steals, blocks)
values (@id, @wk, @fg, @ft, @3fg, @reb, @ast, @stl, @blk)
```

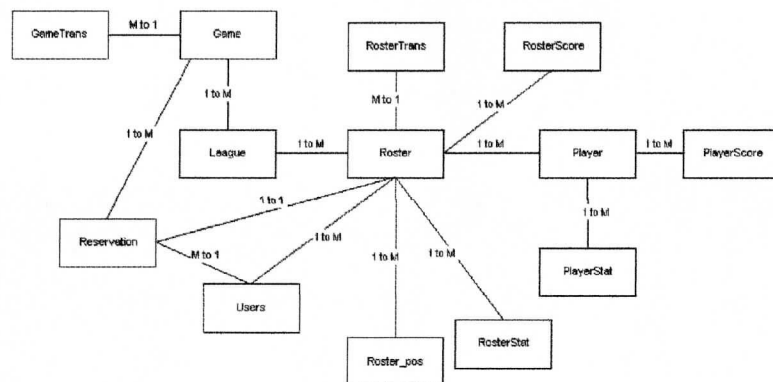
```
Fetch scoringcursor into @id, @fg, @3fg, @ft, @reb, @ast, @stl, @blk
```

```
END
```

## **APPENDIX F**

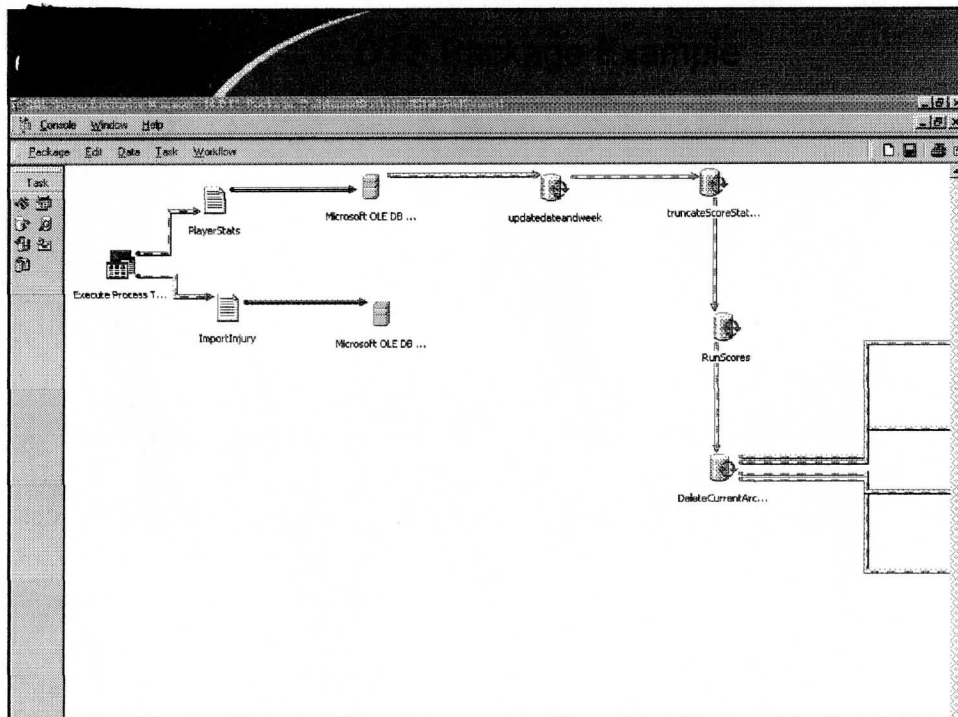


- Database for [REDACTED] basketball game
- Transaction based e-business site
- 8 wk project
- SQL Server 7 dbms
- Cold Fusion Front-end

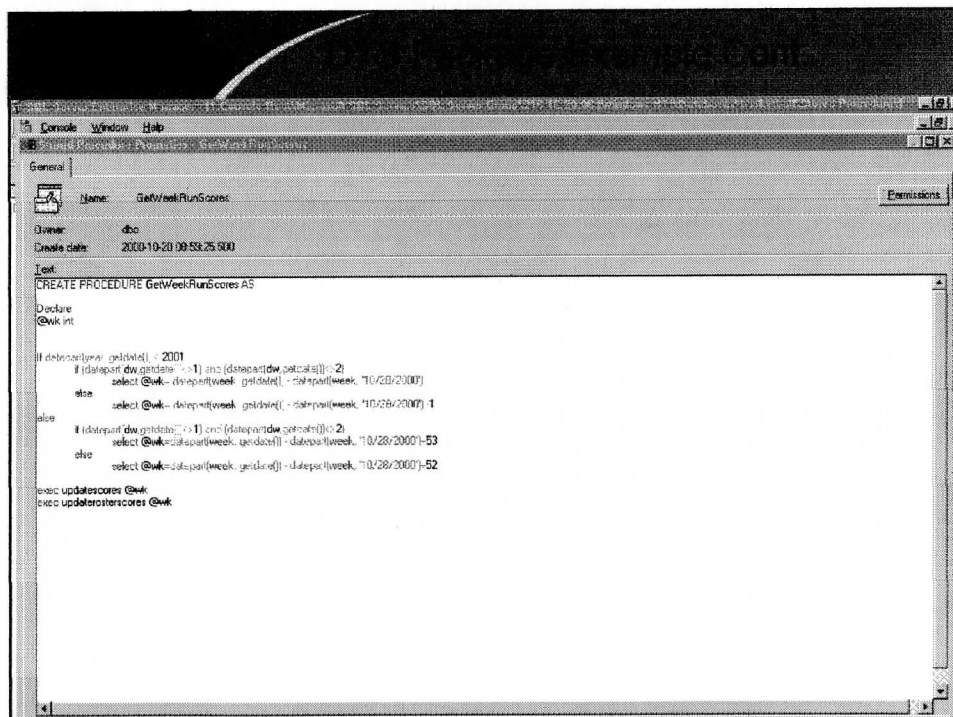
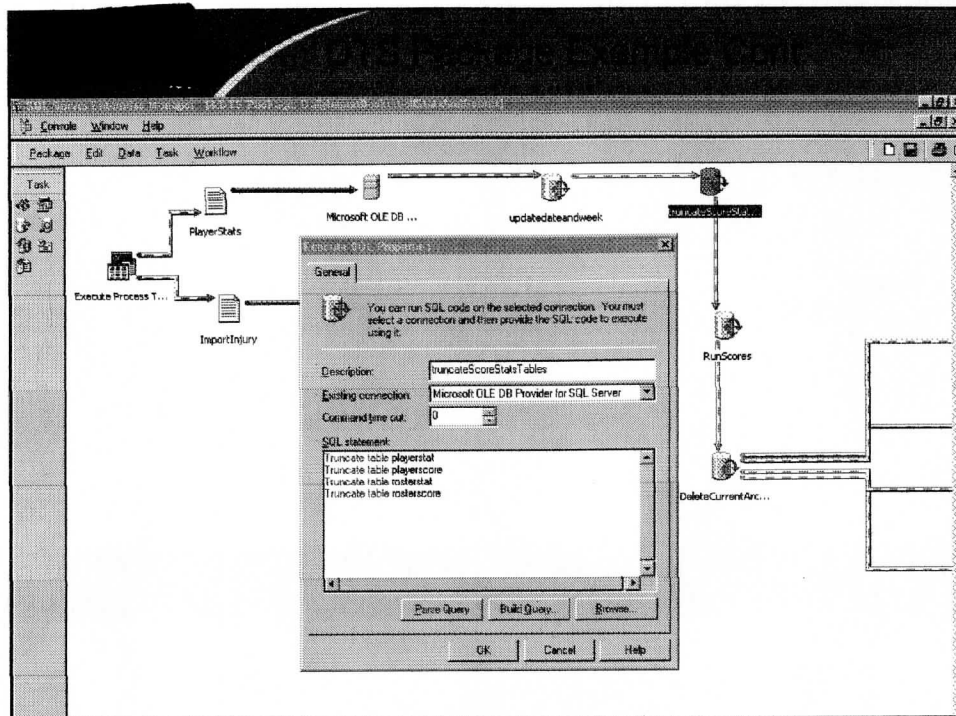


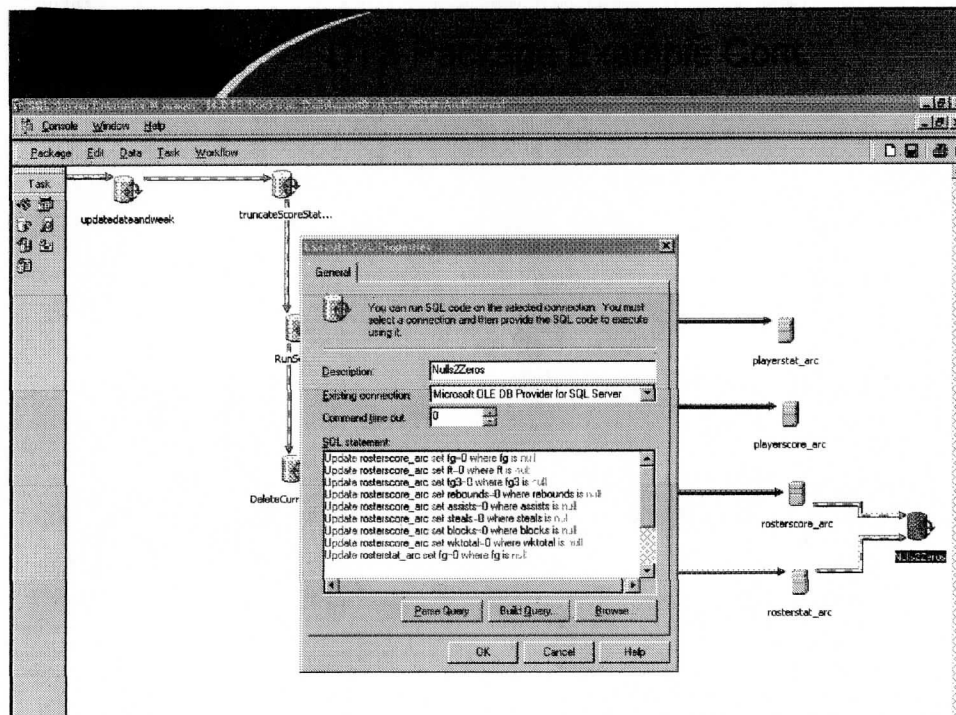
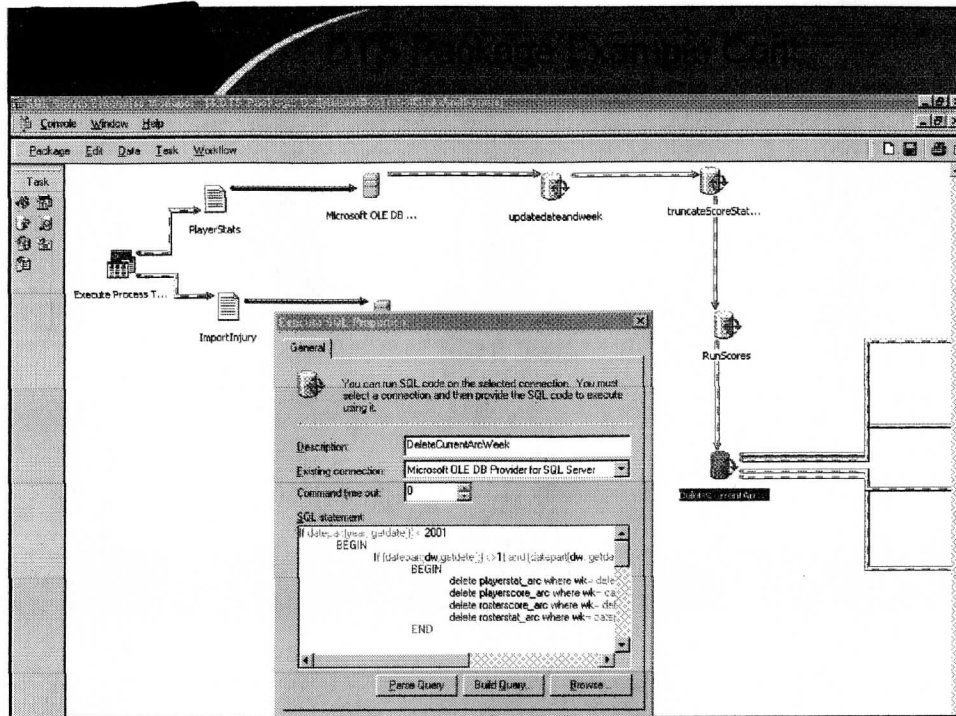


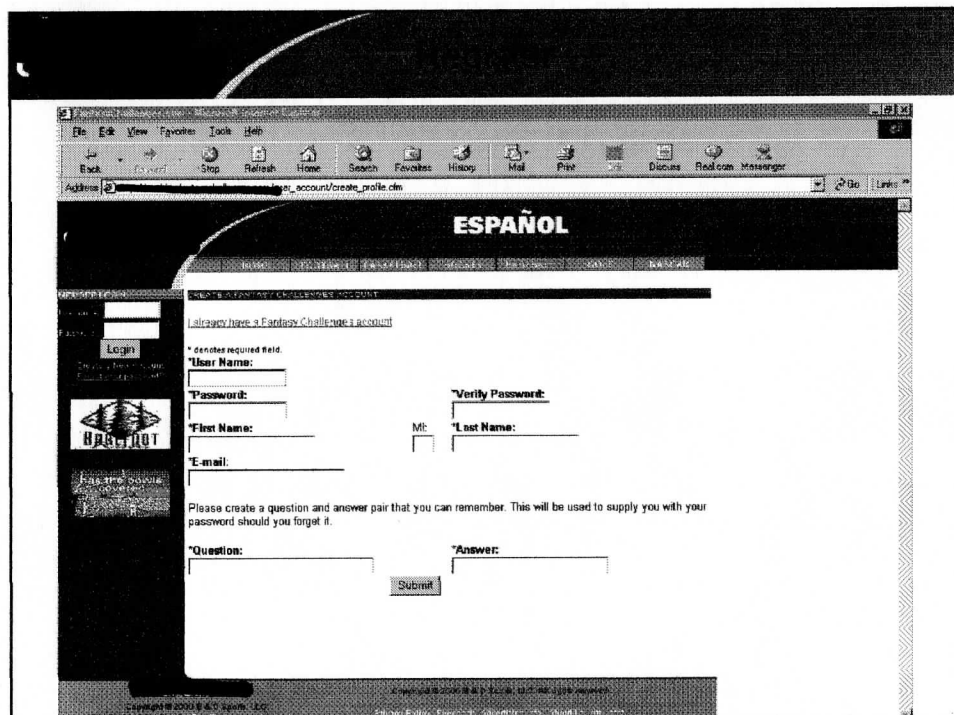
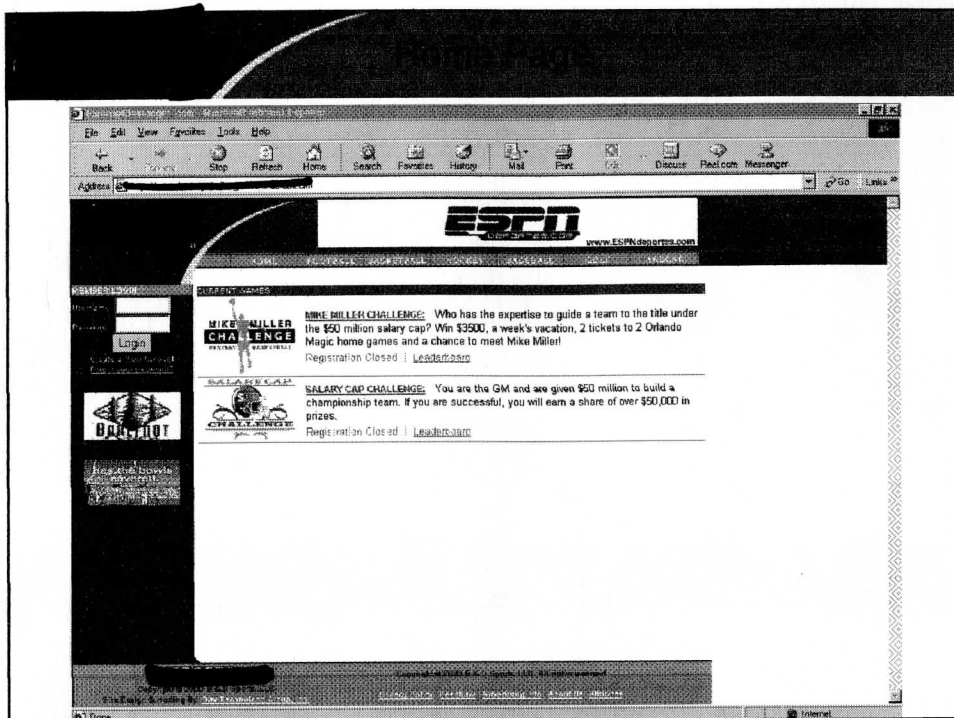
- Player IDs
- Transaction Processing
  - Credit card rejection
  - Exceed trade limit
  - Exceed Salary Cap



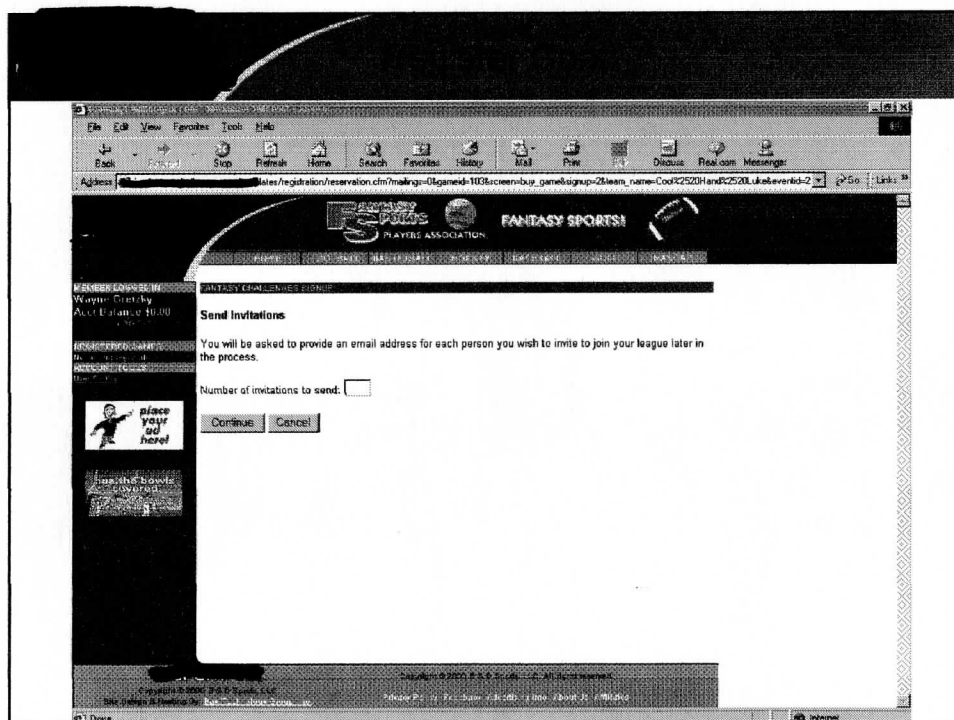
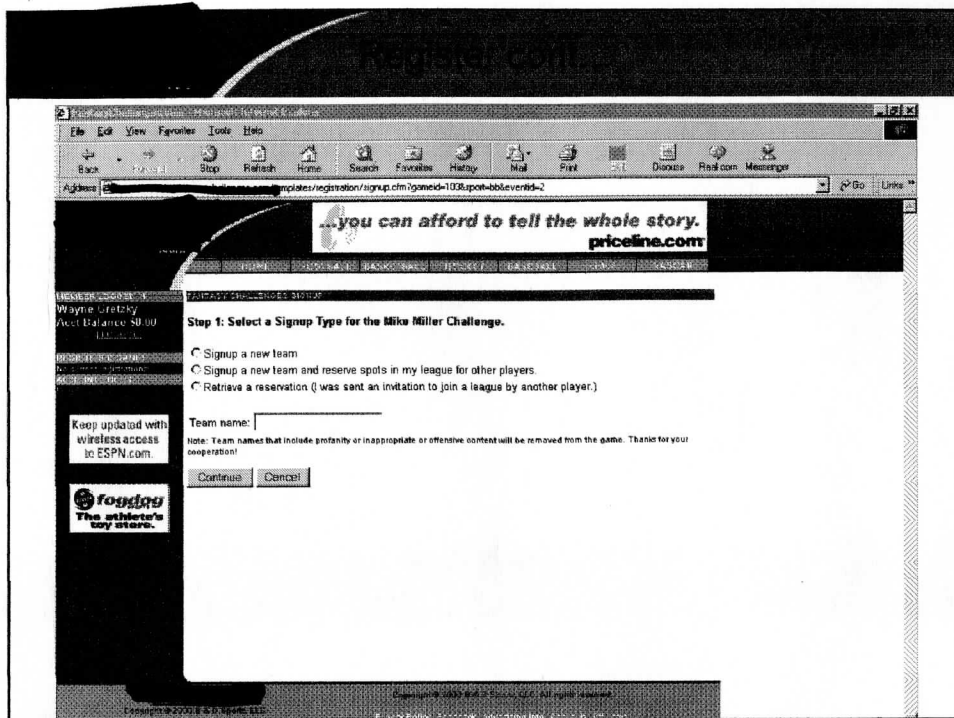




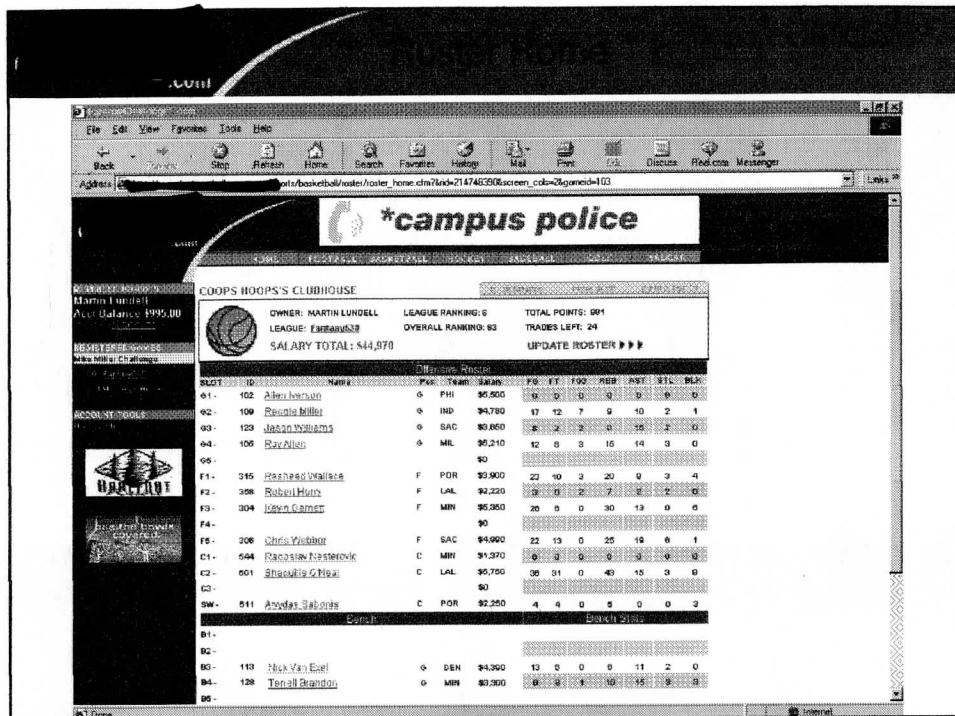












- Sortable/Viewable Stats
- Player Profiles
- Clubhouse Info
  - Trades Remaining
  - Standings
  - Salary Total
  - Total Points

- Roster Moves
  - Add
  - Drop
  - Bench
  - Activate
  - Swing
- Submit Changes
- Credit Card Processing

ESPN

COOPS HOOPS'S CLUBHOUSE

OWNER: MARTIN LUNDALL LEAGUE RANKING: 5 TOTAL POINTS: 1897  
 LEAGUE: Fantasy Hoops OVERALL RANKING: 62 TRADES LEFT: 24  
 SALARY TOTAL: \$49,135 UPDATE ROSTER

YOU ARE UPDATING YOUR ROSTER. CHANGES ARE NOT FINAL UNTIL YOU CLICK SUBMIT CHANGES.  
 CLICK CANCEL TO DISCARD YOUR CHANGES AND RETURN TO THE MAIN ROSTER PAGE.

Submit Changes Cancel

Current Roster				Available Players			
Action	Stat ID	Name	Pos Team Salary	File	Team	Name	Pos Salary
DRP BEN	01	102 Allen Iverson	G PHI \$5,500	101	SEA	Gary Payton	G \$5,500
DRP BEN	02	100 Ray Allen	G MIL \$5,210	103	LAL	Kobe Bryant	G \$5,400
DRP BEN	03	123 Jason Williams	G SAC \$3,000	104	DAL	Michael Finley	G \$5,400
DRP BEN	04	128 Terrell Brandon	G MIN \$3,300	107	MIA	Eddie Jones	G \$5,000
ADD	05			108	DET	Jerome Stachour	G \$5,000
DRP BEN	F1	315 Rashard Wallace	F POR \$3,900	110	KY	Allen Houston	G \$4,620
DRP BEN	F2	308 Robert Horry	F LAL \$2,220	111	HOU	Steve Francis	G \$4,615
DRP BEN	F3	304 Kevin Carroll	F MIN \$5,350	112	WAS	Nate Richmond	G \$4,410
ADD	F4			114	ORL	Corell Annabone	G \$4,300
DRP BEN	F5	300 Chris Webber	F SAC \$4,950	115	PHO	Andreas Karlas	G \$4,310
DRP BEN	C1	603 Dwight Howard	C ATL \$4,875	116	PHO	Jason Kidd	G \$4,125
DRP BEN	C3	501 Shaquille O'Neal	C LAL \$5,750	117	MIL	Sam Cassell	G \$4,000
ADD	C3			118	POR	Steve Smith	G \$3,000
SWAP	SW	113 Nick Van Exel	G DEN \$4,300	119	HOU	Cutino Mobley	G \$3,800
				120	GS	Larry Hughes	G \$3,850
				121	VAN	Mike Dunleavy	G \$3,875
				122	VAN	Michael Dickerson	G \$3,750
				124	CLE	Andra Miller	G \$3,020
				125	MIA	Tim Hardaway	G \$3,500
				126	CHA	Dave Wesley	G \$3,400
				127	ATL	Jim Jackson	G \$3,525
				128	UTA	John Stockton	G \$3,500
				130	CHI	Ron Artest	G \$3,500
				131	SAC	Carmelo Anthony	G \$3,250
				132	POR	Carmelo Anthony	G \$3,210
				133	CLE	Shawn Douglas	G \$3,100
				134	WAS	Bob Stuckard	G \$3,030
				135	SA	Avery Johnson	G \$3,030
				136	CHI	Ron Mercer	G \$3,025

ESPN

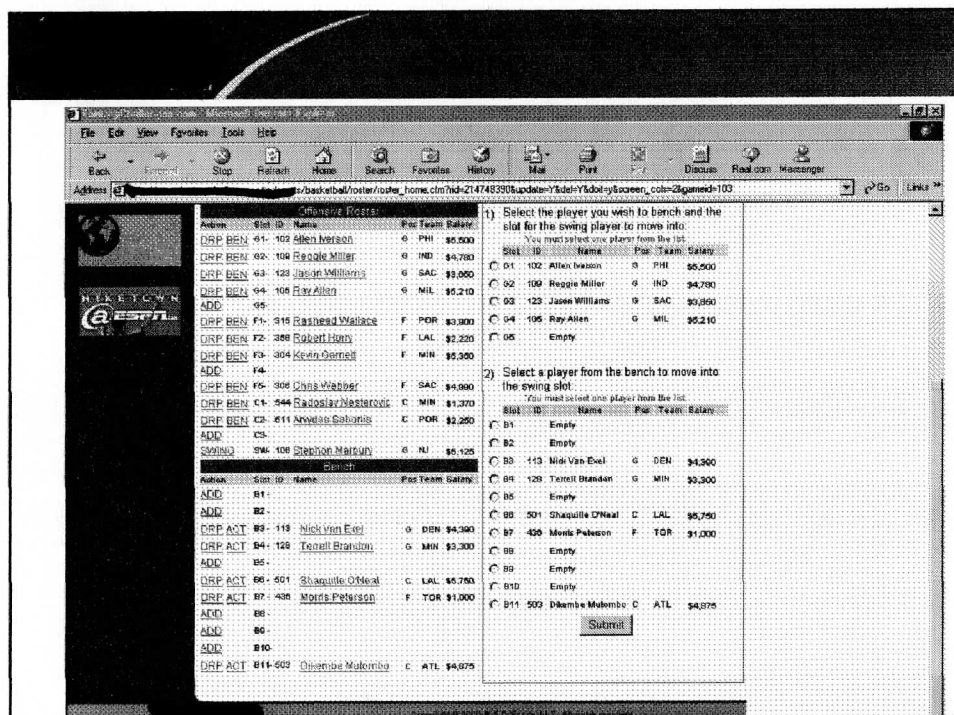
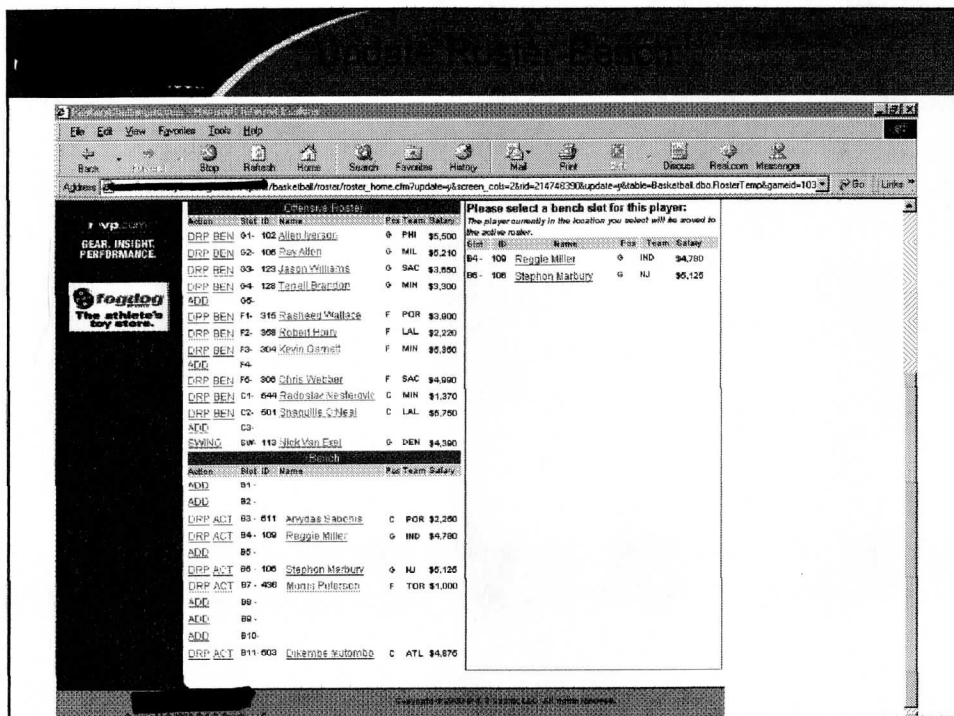
COOPS HOOPS'S CLUBHOUSE

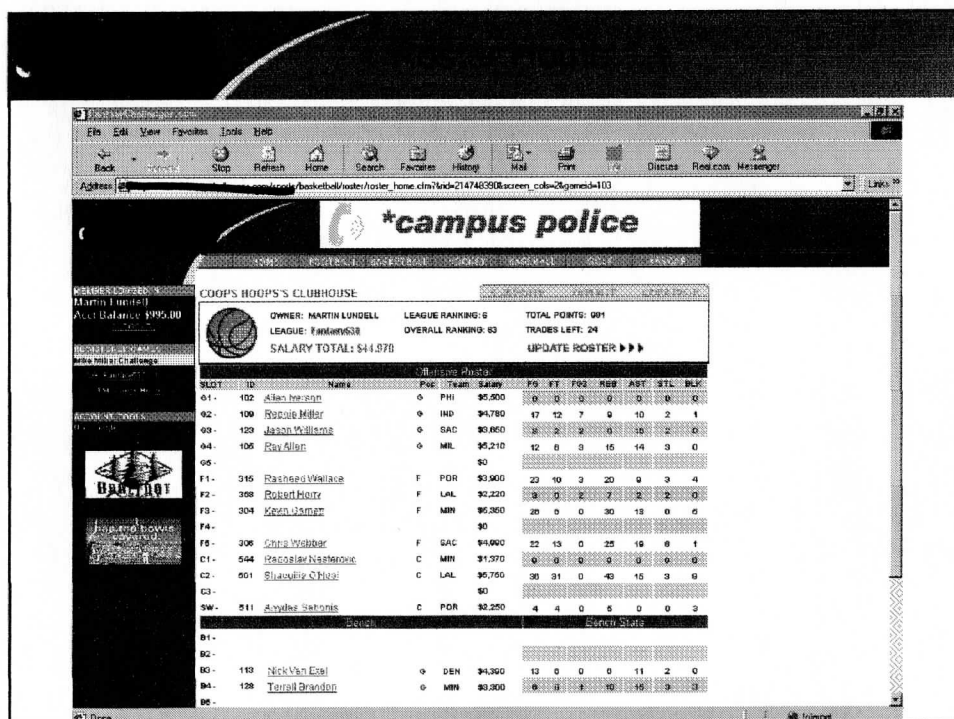
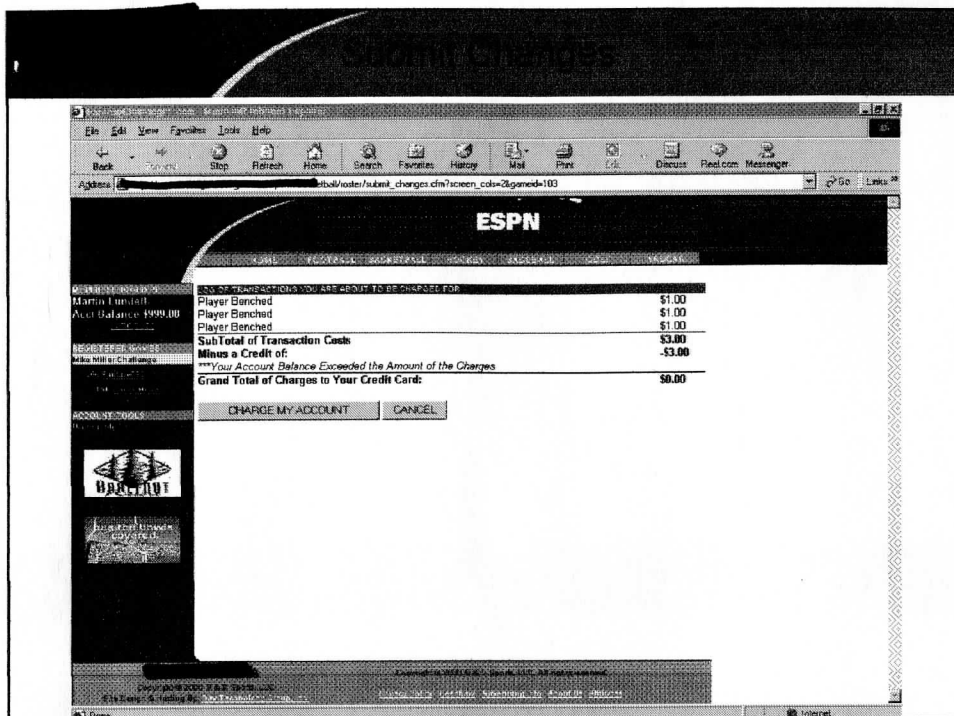
OWNER: MARTIN LUNDALL LEAGUE RANKING: 5 TOTAL POINTS: 1897  
 LEAGUE: Fantasy Hoops OVERALL RANKING: 62 TRADES LEFT: 24  
 SALARY TOTAL: \$49,135 UPDATE ROSTER

YOU ARE UPDATING YOUR ROSTER. CHANGES ARE NOT FINAL UNTIL YOU CLICK SUBMIT CHANGES.  
 CLICK CANCEL TO DISCARD YOUR CHANGES AND RETURN TO THE MAIN ROSTER PAGE.

Submit Changes Cancel

Current Roster				Available Players			
Action	Stat ID	Name	Pos Team Salary	File	Team	Name	Pos Salary
DRP BEN	01	102 Allen Iverson	G PHI \$5,500	101	SEA	Gary Payton	G \$5,500
DRP BEN	02	100 Ray Allen	G MIL \$5,210	103	LAL	Kobe Bryant	G \$5,400
DRP BEN	03	123 Jason Williams	G SAC \$3,000	104	DAL	Michael Finley	G \$5,400
DRP BEN	04	128 Terrell Brandon	G MIN \$3,300	107	MIA	Eddie Jones	G \$5,000
ADD	05			108	DET	Jerome Stachour	G \$5,000
DRP BEN	F1	315 Rashard Wallace	F POR \$3,900	110	KY	Allen Houston	G \$4,620
DRP BEN	F2	308 Robert Horry	F LAL \$2,220	111	HOU	Steve Francis	G \$4,615
DRP BEN	F3	304 Kevin Carroll	F MIN \$5,350	112	WAS	Nate Richmond	G \$4,410
ADD	F4			114	ORL	Corell Annabone	G \$4,300
DRP BEN	F5	300 Chris Webber	F SAC \$4,950	115	PHO	Andreas Karlas	G \$4,310
DRP BEN	C1	603 Dwight Howard	C ATL \$4,875	116	PHO	Jason Kidd	G \$4,125
DRP BEN	C3	501 Shaquille O'Neal	C LAL \$5,750	117	MIL	Sam Cassell	G \$4,000
ADD	C3			118	POR	Steve Smith	G \$3,000
SWAP	SW	113 Nick Van Exel	G DEN \$4,300	119	HOU	Cutino Mobley	G \$3,800
				120	GS	Larry Hughes	G \$3,850
				121	VAN	Mike Dunleavy	G \$3,875
				122	VAN	Michael Dickerson	G \$3,750
				124	CLE	Andra Miller	G \$3,020
				125	MIA	Tim Hardaway	G \$3,500
				126	CHA	Dave Wesley	G \$3,400
				127	ATL	Jim Jackson	G \$3,525
				128	UTA	John Stockton	G \$3,500
				130	CHI	Ron Artest	G \$3,500
				131	SAC	Carmelo Anthony	G \$3,250
				132	POR	Carmelo Anthony	G \$3,210
				133	CLE	Shawn Douglas	G \$3,100
				134	WAS	Bob Stuckard	G \$3,030
				135	SA	Avery Johnson	G \$3,030
				136	CHI	Ron Mercer	G \$3,025





- Overall Standings
- League Standings
- Viewable Rosters
- Sortable Stats

Internet Explorer - \*donkey

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Discuss Real.com Messages

Address http://baseballleague.com/league\_home.cfm?&ipwentyfive=7186-31185405301table=roster&screen\_edit=2147483647&gameid=103

**\*donkey**

COOPS HOOPS'S CLUBHOUSE

OWNER: MARTIN LUNDELL LEAGUE RANKING: 6 TOTAL POINTS: 1907  
 LEAGUE: Fantasy538 OVERALL RANKING: 62 TRADES LEFT: 24  
 SALARY TOTAL: \$43,060 UPDATE ROSTER >>>

LEAGUE STANDINGS

WEEK 1 2 3 4 5 6 7 8 9 10 11 12 13

14 15 16 17 18 19 20 21 22 23 24 N/A

OVERALL STANDINGS

Place	Team Name	PP	FT	3PT	REB	AST	STL	BLK	PTS
1	Navy BB-2	8378	5206	1044	10812	2803	2460	2541	34364
2	Navy BB-2	9400	5206	1047	10790	2805	2476	2525	34328
3	WARRIORS	9072	5312	1239	10088	2819	2400	2448	33236
4	FLORIDA LUMBERJACKS	8544	5464	1251	9972	2460	2214	2151	32675
5	FLORIDA LUMBERJACKS	8794	5414	1032	9870	2464	2343	2709	32010
6	Blood Mary's II	9244	5290	675	9896	2279	2430	2505	32059
7	FLORIDA LUMBERJACKS	8880	5100	861	10040	2531	2203	2037	32028
8	FLORIDA LUMBERJACKS	9104	5206	1113	9818	2590	2418	2202	32481
9	FLORIDA LUMBERJACKS	8652	4940	1125	10014	2684	2310	2379	32408
10	FLORIDA LUMBERJACKS	8038	5462	1118	9582	2283	2236	2544	32288
11	The Bulls	8612	5306	1129	9590	2368	2460	2555	32168
12	SALARY PAIN	8724	4912	1025	9546	2170	2100	2446	31668
13	FLORIDA LUMBERJACKS	8564	4880	1038	9874	2451	2305	2220	31602
14	DEATH SQUAD	8948	4900	1032	9590	2743	2142	2100	31023
15	FLORIDA LUMBERJACKS	8654	5264	1254	9806	2331	2187	1977	30803
16	FLORIDA LUMBERJACKS	8686	4988	1443	9186	2708	2408	2100	30603
17	FLORIDA LUMBERJACKS	8204	5104	1199	9090	2182	2271	2100	30100
18	FLORIDA LUMBERJACKS	8178	4464	627	9308	3337	2253	2031	29990
19	FLORIDA LUMBERJACKS	8972	4722	1020	8468	2517	2322	2124	29881
20	FLORIDA LUMBERJACKS	7774	4900	795	8304	2358	2043	2523	28738
21	FLORIDA LUMBERJACKS	9302	4672	1283	8730	2409	2073	2124	28673
22	FLORIDA LUMBERJACKS	8398	5000	1289	8590	2195	2064	2022	28603

Shopify  
The Point 'N' the Point  
foxdog  
The athlete's  
toy store.



Internet Explorer 5.0

Address: http://sports.basketball/leagues/league\_home.cfm?id=201654850&id=214748350&screen\_cals=2&gameid=103&leagueable=1&scores\_id=1

**COOPS HOOPS'S CLUBHOUSE**

OWNER: MARTIN LUNDRELL  
LEAGUE: Fantasy538  
SALARY TOTAL: \$43,060

LEAGUE RANKING: 5  
OVERALL RANKING: 82

TOTAL POINTS: 1007  
TRADES LEFT: 24  
UPDATE ROSTER >>>

**LEAGUE STANDINGS**

League: Fantasy538

WEEK	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	1	2	3	4	5	6	7	8	9	10	11	12	13	14
4	1	2	3	4	5	6	7	8	9	10	11	12	13	14
5	1	2	3	4	5	6	7	8	9	10	11	12	13	14
6	1	2	3	4	5	6	7	8	9	10	11	12	13	14
7	1	2	3	4	5	6	7	8	9	10	11	12	13	14
8	1	2	3	4	5	6	7	8	9	10	11	12	13	14
9	1	2	3	4	5	6	7	8	9	10	11	12	13	14
10	1	2	3	4	5	6	7	8	9	10	11	12	13	14

**Overall Standings**

Rank	Team Name	FG	FT	3P	REB	AST	STL	BLK	PTS
1	The Pit Bulls	8750	4874	900	8072	2229	2244	2127	20889
2	Hot Life	8120	4062	1002	8184	1633	2118	2181	20290
3	Stitchers	7554	4034	831	6774	2401	2070	1896	27060
4	Crooks	7414	4220	1017	7412	1742	1752	1920	25402
5	Coops Hoops	6188	3370	1203	6638	1920	1690	1300	21533
6	Shannon	5786	3222	1041	4704	1852	1691	994	19060
7	Adrenaline Whores	4040	2214	806	2455	1220	1227	432	12145
8	Mr. Martin	1688	706	108	1836	358	372	466	8523
9	Stitchers2	004	294	132	474	294	288	54	2180
10	King Joke	0	0	0	0	0	0	0	0

Internet Explorer 5.0

Address: http://sports.basketball/detail/roster\_view.cfm?rosterid=201654850&id=214748350&gameid=103

**ESPN OL**

**SHRICHIVES'S CLUBHOUSE**

OWNER: BRENT STRICHERZ  
LEAGUE: Fantasy538  
SALARY TOTAL: \$49,455

LEAGUE RANKING: 2  
OVERALL RANKING: 31

TOTAL POINTS: 1363  
TRADES LEFT: 28

**Active Roster for Week 20**

SLOT	ID	Name	Pos	Team	Salary	FG	FT	3P	REB	AST	STL	BLK
01	120	Terrill Brandon	G	MIN	\$5,300	18	4	2	12	30	4	1
02	136	David Best	G	IND	\$2,800	14	3	4	7	21	7	0
03	195	Mark Jackson	G	TOR	\$2,300	10	2	3	14	21	2	0
04	101	Ray Parker	G	SBA	\$5,550	26	10	10	10	22	8	0
05	111	Steve Francis	G	HOU	\$4,815	18	14	3	10	27	4	1
F1	303	Karl Malone	F	UTA	\$5,400	20	15	0	20	12	1	1
F2	372	Tim Thomas	F	MIL	\$1,800	20	12	12	17	7	7	3
F3	307	Shane'Andre' Patten	F	VAN	\$4,950	18	24	1	20	10	5	3
F4	378	Jeanine O'Neal	F	IND	\$1,800	20	18	0	50	6	3	14
F6	240	Glenn Davis	F	MIA	\$2,340	25	9	0	15	4	5	0
C1	544	Francis Nwankwo	C	MIN	\$1,370	2	0	0	5	0	0	0
C2	501	Shaquille O'Neal	C	LAL	\$5,700	40	25	0	40	12	2	8
C3	526	Marques Kennedy	C	NY	\$1,770	10	4	0	21	0	5	0
SW	304	Brent Campbell	F	MIN	\$5,350	27	23	0	52	21	5	3

Internet Explorer - Coops Hoops's Clubhouse

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Discuss Real.com Messenger

Address: http://sports.basketball.coops/hoops\_home.cfm?id=214748390&screen\_code=26&gameid=103&do=ny

**COOPS HOOPS'S CLUBHOUSE**

INJURY REPORT  
PLAYER RESEARCH  
NEA GAME SCHEDULE

TRANSACTION LOG

SLOT	ID	Name	Pos	Team	Salary	PG	FT	P43	REB	AST	STL	BLK
01-	102	Allen Iverson	G	PHI	\$5,900	27	19	5	7	5	2	0
02-	100	Reggie Miller	G	IND	\$4,780	4	3	1	2	2	0	0
03-	123	Jason Williams	G	SAC	\$3,060	7	0	1	0	5	2	0
04-	105	Ray Allen	G	MIL	\$3,210	7	1	1	1	6	0	0
05-					\$0							
F1-	315	Rasheed Wallace	F	POR	\$3,900	5	1	1	5	3	0	4
F2-	389	Robert Horn	F	LAL	\$2,220	0	0	0	0	0	0	0
F3-	304	Kevin Garnett	F	MIN	\$5,350	7	2	1	14	8	1	7
F4-					\$0							
F5-	309	Chris Webber	F	SAC	\$4,900	13	5	0	11	5	1	1
C1-	544	Radoslaw Nasternycki	C	MIN	\$1,370	0	0	0	0	0	0	0
C2-	511	Arvydas Sabonis	C	POR	\$2,260	4	2	0	4	3	1	2
C3-					\$0							
SW-	100	Stephon Marbury	G	NY	\$3,125							

Back CAPCH/CR/AL

Internet Explorer - Coops Hoops's Clubhouse

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Discuss Real.com Messenger

Address: http://basketball.coops/hoops\_home.cfm?id=214748390&screen\_code=26&gameid=103

**COOPS HOOPS'S CLUBHOUSE**

OWNER: MARTIN LUNDELL  
LEAGUE: Fantasy930  
SALARY TOTAL: \$44,970

LEAGUE RANKING: 9  
OVERALL RANKING: 63

TOTAL FORTS: 961  
TRADES LEFT: 24

UPDATE ROSTER >>>

**INJURED PLAYERS ON YOUR ROSTER**

Last Name	First Name	Team	Position	Date	Description
Robert	Horry	LAL	PF	03/23/01	Cervical strain
Allen	Iverson	Phi	SG	03/14/01	left hip and flank contusion
Nick	Van Exel	Den	PG	03/05/01	Left knee contusion
Shaquille	O'Neal	LAL	C	01/28/01	Sore right foot
Allen	Iverson	Phi	SG	01/19/01	Left knee contusion
Allen	Iverson	Phi	SG	12/25/00	Dislocated right shoulder
Rasheed	Wallace	POR	F-C	12/09/00	DNP 12/9 - some right wrist
Stephon	Marbury	NJ	PG	12/05/00	Sprained Left Ankle
Arvydas	Sabonis	POR	C	12/02/00	Left Knee Sprain
Arvydas	Sabonis	POR	C	11/14/00	no longer on IL
Terrell	Brandon	Min	PG	11/11/00	Sprained left ankle
Dikembe	Mutombo	Atl	C	11/03/00	no longer on IL

**FULL INJURY REPORT**

Last Name	First Name	Team	Position	Date	Description
Curry	Michael	Det	SF	03/24/01	dislocated right thumb
David	Kornet	Tor	PF	03/24/01	no longer on IL
Bryant	Kobe	LAL	G-F	03/23/01	sore left foot
Harris	Lucious	NJ	G	03/23/01	DNP 3/23 - personal reasons
Douglas	Sherman	PG	PG	03/23/01	fractured left thumb
Rice	Glen	NY	G-F	03/23/01	plantar fasciitis left foot
Martin	Kenyon	NJ	PF	03/23/01	fractured right leg
Horry	Robert	LAL	PF	03/23/01	Cervical strain
Camby	Marcus	NY	F-C	03/23/01	upper respiratory infection
Lue	Tyrone	LAL	PG	03/23/01	sprained left ankle

Internet Explorer window showing a fantasy basketball league page titled "COOPS HOOPS'S CLUBHOUSE". The page displays player statistics, team information, and a weekly player ratings table.

**COOPS HOOPS'S CLUBHOUSE**

OWNER: MARTIN LUNDSELL  
LEAGUE: Fantasy  
SALARY TOTAL: \$44,970

LEAGUE RANKING: 8  
OVERALL RANKING: 83  
TOTAL POINTS: 1001  
TRADES LEFT: 24  
UPDATE ROSTER

**PLAYER: Allen Iverson**  
**SALARY: \$5,500**  
**TEAM: 76ers**  
**POSITION: Guard**

**WEEKLY PLAYER RATINGS**

WEEK	FG	FT	3FG	REB	STL	BLK
1	32	34	3	26	7	2
2	22	16	3	6	3	1
3	17	7	0	10	4	0
4	27	19	3	16	11	2
5	31	25	6	23	9	0
6	34	42	2	17	9	2
7	38	33	1	14	10	1
8	20	23	1	3	6	0
9	31	9	4	12	5	0
10	42	25	7	8	7	2
11	39	23	2	11	8	1
12	46	29	8	7	10	1
13	38	53	4	12	15	2
14	49	40	10	20	8	2
15	24	13	2	10	3	0
16	56	35	10	13	12	1
17	43	35	5	14	7	1
18	29	30	6	7	6	0
19	32	29	3	9	11	0
20	11	4	2	4	2	0
21	0	0	0	0	0	0

Internet Explorer window showing a fantasy basketball league page titled "COOPS HOOPS'S CLUBHOUSE". The page displays player statistics, team information, and a weekly player ratings table.

**COOPS HOOPS'S CLUBHOUSE**

OWNER: MARTIN LUNDSELL  
LEAGUE: Fantasy  
SALARY TOTAL: \$44,970

LEAGUE RANKING: 8  
OVERALL RANKING: 83  
TOTAL POINTS: 1001  
TRADES LEFT: 24  
UPDATE ROSTER

**COMPLETE TRANSACTION LOG**

Click on column headings to sort.

Date	Time	Action	Player In	Player Out
02/14/2001	4:12 PM	Player Bench	Terrell Brandon	Ray Allen
02/05/2001	7:45 PM	Player Bench	Nick Van Exel	Anydas Sabonis
02/05/2001	7:43 PM	Player Bench	Ray Allen	Reggie Miller
02/05/2001	7:41 PM	Player Bench	Dikembe Mutombo	Radoslaw Nesterowicz
01/05/2001	11:35 AM	Player Added		Shaquille O'Neal
12/26/2000	10:17 PM	Player Bench	Stephon Marbury	Terrell Brandon
12/26/2000	10:16 PM	Player Added	Robert Horry	
11/01/2000	2:39 PM	Player Added	Rasheed Wallace	
11/01/2000	1:55 PM	Player Added	Stephon Marbury	
11/01/2000	1:49 PM	Player Added	Jason Williams	
11/01/2000	1:48 PM	Player Added	John Stockton	
10/24/2000	6:31 PM	Player Bench - Preseason	Reggie Miller	Ray Allen
10/20/2000	3:40 PM	Player Added - Preseason	Monte Peterson	
10/20/2000	3:39 PM	Player Added - Preseason	Terrell Brandon	
10/20/2000	1:30 PM	Player Added - Preseason	Shaquille O'Neal	
10/20/2000	1:29 PM	Player Added - Preseason	Ray Allen	
10/17/2000	12:11 PM	Player Swing - Preseason	Nick Van Exel	Radoslaw Nesterowicz
10/16/2000	10:10 AM	Player Bench - Preseason	Nick Van Exel	Dikembe Mutombo
10/16/2000	10:09 AM	Player Swing - Preseason	Nick Van Exel	
10/13/2000	2:22 PM	Player Swing - Preseason	Kevin Gammett	Anydas Sabonis
10/13/2000	1:34 PM	Player Added - Preseason	Anydas Sabonis	
10/11/2000	4:16 PM	Player Dropped		John Stockton



Click here to find out. **at the KELOLAND Mall.** KELOLAND MALL

**KELOLAND.COM**  
Fantasy Home

**COOPS HOOPS'S CLUBHOUSE**

OWNER: MARTIN LUNDELL LEAGUE RANKING: 6 TOTAL POINTS: 1127  
LEAGUE: Fantasy2000 OVERALL RANKING: 63 TRADES LEFT: 24  
SALARY TOTAL: \$44,345 UPDATE ROSTER >>>

SLCT	PT	NAME	POS	TEAM	SALARY	PPG	FT	FG	REB	AST	STL	BLK
01-	100	Allen Iverson	G	PHI	\$2,500	7	3	1	1	2	0	0
02-	100	Reggie Miller	G	IND	\$4,780	19	14	4	8	13	3	1
03-	123	Jason Williams	G	SAC	\$3,000	0	3	0	30	27	4	0
04-	100	Ronny Olin	G	MIL	\$5,210	31	17	9	11	12	4	0
05-					\$0							
F1-	315	Francisco M. Williams	F	POR	\$3,900	13	5	0	27	11	3	7
F2-	350	Robert Horry	F	LAL	\$2,220	7	2	4	16	9	4	1
F3-	304	Kevin Garnett	F	MIN	\$5,350	21	11	1	29	13	4	7
F4-					\$0							
F5-	305	Chris Webber	F	SAC	\$4,990	17	10	0	33	14	5	2
C1-	691	Reggie Miller	C	MIN	\$1,370	0	0	0	0	0	0	0
C2-	611	Vladimir Stepanovich	C	POR	\$2,280	28	12	0	30	5	5	0
C3-					\$0							
SW-	100	Stephen Marbury	G	NY	\$5,120	0	0	0	0	0	0	0
B1-												
B2-												

CREATE USER PROFILE

User Name:  Password:

Fields marked \* are required

\*First Name:  MI:  \*Last Name:

\*Address 1:

Address 2:

\*City:  \*State/Province:

\*Zip Code:  \*Country:

\*Phone Number (xxx-xxx-xxxx):  \*E-mail:

Credit Card:

Visa ☒ MasterCard ☐ American Express ☐

Name as Appears on Credit Card:  Expiration Date (mm/yy):

Credit Card Number (xxxx-xxxx-xxxx-xxxx):

☐ Hide my user profile from other players.

Create User Profile